

*Um Algoritmo para Abdução Peirceana Baseado em uma  
Estrutura de Raciocínio Quaternária em Lógica  
Proposicional*

**Felipe Rodrigues**

Abril / 2015

Dissertação de Mestrado em Ciência da  
Computação

# **Um Algoritmo para Abdução Peirceana Baseado em uma Estrutura de Raciocínio Quaternária em Lógica Proposicional**

Esse documento corresponde à dissertação apresentada à Banca Examinadora no curso de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista.

Campo Limpo Paulista, 7 de Abril de 2015.

Felipe Rodrigues

Oswaldo Luiz de Oliveira (Orientador)

**Faculdade Campo Limpo Paulista**  
**Programa de Mestrado em Ciência da Computação**

*Um Algoritmo para Abdução Peirceana Baseado em uma Estrutura de Raciocínio Quaternária em Lógica Proposicional*

FELIPE RODRIGUES

*Dissertação de Mestrado apresentado ao Programa de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.*

Membros da Banca:



Prof. Dr. Osvaldo Luiz de Oliveira

(Orientador – FACCAMP)



Profa. Dra. Ana Maria Monteiro

(FACCAMP)



Profa. Dra. Maria do Carmo Nicoletti

(FACCAMP)



Profa. Dra. Maria Cecilia Calani Baranauskas

(UNICAMP)

Campo Limpo Paulista, 07 de abril de 2015.

## FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Câmara Brasileira do Livro, São Paulo, Brasil

Rodrigues, Felipe

Um algoritmo para abdução peirceana baseado em uma estrutura de raciocínio quaternária em lógica proposicional / Felipe Rodrigues. Campo Limpo Paulista, SP: FACCAMP, 2015.

Orientador: Prof<sup>o</sup>. Dr. Osvaldo Luiz de Oliveira

Dissertação (mestrado) – Faculdade Campo Limpo Paulista – FACCAMP.

1. Raciocínio abduutivo. 2. Raciocínio automatizado. 3. Lógica. 4. Fatores humanos. 5. Inteligência artificial. I. Oliveira, Osvaldo Luiz de. II. Faculdade Campo Limpo Paulista. III. Título.

CDD-005.1

**Resumo:** *Raciocínios abdutivos formulam hipóteses para explicar fatos observados, considerando uma teoria como base. Inúmeras tarefas intelectuais incluindo diagnóstico médico, diagnóstico de falhas, descoberta científica, argumentação jurídica e interpretação de linguagem natural fazem uso de raciocínio abductivo. Raciocínios dedutivos conduzem a conclusões irrefutáveis. Diferentemente, os raciocínios abdutivos estabelecem possíveis hipóteses explicativas de fatos observados, algumas melhores e outras piores. Algoritmos para raciocínio abductivo com referencial em lógica têm sido propostos. Entretanto, eles têm de forma geral as seguintes desvantagens: (1) não explicitam como levar em conta condições que podem afetar o raciocínio tais como contexto, circunstância, intenção, crenças, fé e credências; (2) dão pouca ou nenhuma atenção aos critérios de seleção de boas hipóteses. Tendo como ferramenta a Lógica Proposicional, esta dissertação de mestrado propõe o desenvolvimento e implementação de um algoritmo, denominado Peirce, para formulação de hipóteses abductivas com base em uma estrutura de raciocínio quaternária que permite explicitar a inclusão de condições para a realização de raciocínios abdutivos e que leva em consideração critérios para a seleção de boas hipóteses.*

**Palavras-chave:** raciocínio abductivo, raciocínio automatizado, lógica, fatores humanos, inteligência artificial.

**Abstract:** *Abductive reasoning formulate hypotheses to explain observed facts, considering a theory as base. Numerous intellectual tasks like medical diagnosis, fault diagnosis, scientific research, legal reasoning and natural language interpretation make use of abductive reasoning. Deductive reasoning lead to irrefutable conclusions. Opposite to that, abductive reasoning establishes possible explainable hypothesis to observed facts, some better than others. Logic based algorithms have been proposed to perform abductive reasoning. However, they usually have the following disadvantages: (1) they do not explain how to account for conditions that might affect the abductive reasoning, conditions like context, circumstances, intentions, faith, beliefs, etc.; (2) they tend to give little to no attention to the criteria of selection of good hypothesis. Using Propositional Logic, this master's thesis proposes the development and implementation of an algorithm, named Peirce, which formulates abductive hypothesis based on a quaternary reasoning structure that allows the explicit inclusion of conditions when performing abductive reasoning and accounts for a criteria for the selection of good hypotheses.*

**Keywords:** abductive reasoning, automated reasoning, logic, human factors, artificial intelligence.

# Sumário

Capítulo 1 Introdução.....	1
1.1 – Tipos de raciocínio.....	2
1.2 – Abdução e indução .....	5
1.3 – Exemplos de uso de raciocínio abduativo .....	7
Cenário 1.1 – Situação do cotidiano.....	8
Cenário 1.2 – Situação do cotidiano.....	8
Cenário 1.3 – Diagnóstico médico.....	9
Cenário 1.4 – Descoberta de falhas em sistemas com comportamento conhecido.....	9
Cenário 1.5 – Intencionalidade no raciocínio abduativo – acusação e defesa .....	9
Cenário 1.6 – Descoberta científica .....	10
1.4 – Critérios para a seleção de boas hipóteses .....	11
1.5 – Organização da Dissertação .....	13
Capítulo 2 Contexto de Estudo, Objetivo e Metodologia .....	14
2.1 – Lógica Proposicional .....	15
2.2 – A Lógica como formalismo para o raciocínio abduativo .....	25
2.3 – Abordagens em Lógica para realização do raciocínio abduativo .....	28
2.4 – Desvantagens das abordagens existentes.....	29
2.5 – Objetivo .....	30
2.6 – Metodologia .....	30
Capítulo 3 Estrutura Quaternária TCHF e um Critério para Seleção de Boas Hipóteses .....	33
3.1 – Condições Aceitas.....	34
3.2 – Estrutura Quaternária TCHF .....	36
3.3 – Critério para a Seleção de Boas Hipóteses.....	39
Capítulo 4 Algoritmo Peirce em Lógica Proposicional .....	44
4.1 – Algoritmo Peirce em Lógica Proposicional .....	45
4.2 – Detalhamento do algoritmo Peirce .....	49
4.2.1 – Consistentes(A, B) .....	49
4.2.2 – Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C) .....	50
4.2.3 – Resolucao(R).....	52
4.2.4 – Formular_Hipoteses_Candidatas(R).....	55
4.2.5 – Remover_Hipoteses_Inconsistentes(T, C, H) .....	56
4.2.6 – Selecionar_Boas_Hipoteses(T, C, H, F).....	60
4.3 – Tempo de execução.....	63

4.4 – Pesquisa Experimental .....	64
4.4.1 – Hipóteses de pesquisa.....	64
4.4.2 – Materiais.....	64
4.4.3 – Sujeitos da pesquisa .....	65
4.4.4 – Procedimentos experimentais .....	65
4.4.5 – Resultados e discussão.....	66
Capítulo 5 Peirce Online .....	68
5.1 – Peirce Online .....	69
5.2 – Arquitetura do Sistema Peirce Online.....	76
5.3 – Gramática da Linguagem Pierce Abdutiva .....	77
5.4 – O Compilador da Linguagem Peirce Abdutiva.....	79
Capítulo 6 Discussão e Conclusão .....	80
6.1 – Estrutura TCHF.....	81
6.2 – Critério para a Seleção de Boas Hipóteses.....	82
6.3 – Algoritmo Peirce.....	83
6.4 – Linguagem Peirce Abdutiva .....	85
6.5 – O Sistema Peirce Online .....	85
6.6 – Trabalhos Futuros.....	89
Apêndice I Mais cenários exemplo do uso do raciocínio abduativo.....	91
Cenário I.1 – Investigação policial (adaptado de um caso real).....	91
Cenário I.2 – Interpretação de texto .....	92
Cenário I.3 – Interpretação de pinturas e imagens.....	93
Apêndice II Algoritmos para Abdução baseados em Lógica .....	94
II.1 – Tableaux Semântico .....	94
II.2 – Tableaux Semântico para abdução .....	96
II.3 – Algoritmo para raciocínios abduativos de Aliseda baseado em Tableaux Semântico .....	98
II.4 – Programação Lógica Abdutiva (ALP) .....	101
Apêndice III Questionário.....	103
Gabarito do Questionário (alternativas computadas pelo algoritmo Peirce).....	112
Apêndice IV Pseudocódigo do compilador da linguagem Peirce Abdutiva .....	113
Apêndice V Artigo Publicado.....	121
Referências.....	136

# Capítulo 1

## Introdução

*Este capítulo descreve detalhadamente o objeto de estudo deste trabalho, o raciocínio abduativo. Ele inicia discutindo os tipos de raciocínio segundo uma interpretação da concepção do filósofo americano Charles Sanders Peirce segundo a qual existem três tipos de raciocínio: o dedutivo, o indutivo e o abduativo. A Seção 1.1 apresenta a interpretação dotada neste trabalho para o significado destes três tipos de raciocínio, tomada a partir de pequenos fragmentos de textos escritos por Peirce e de certo consenso entre pesquisadores contemporâneos. Considerando que raciocínios abduativos e indutivos são, muitas das vezes, tratados como um único tipo de raciocínio e, outras vezes, como um sendo um subtipo do outro, a Seção 1.2 analisa as diferenças e similaridades entre estes dois tipos de raciocínio, segundo a perspectiva adotada neste trabalho. A Seção 1.3 apresenta diversos exemplos de emprego de raciocínio abduativo como motivação ao estudo desse raciocínio. Raciocínios abduativos formulam possíveis hipóteses para explicar fatos observados. Uma vez que algumas hipóteses podem explicar melhor os fatos observados do que outras, raciocínios abduativos podem ser vistos como raciocínios para boas hipóteses, assunto tratado na Seção 1.4 e, por fim a Seção 1.5 descreve a organização desta dissertação.*

## 1.1 – Tipos de raciocínio

De acordo com filósofo Charles Sanders Peirce existem três tipos de raciocínio: dedução, indução e abdução (Peirce, 1867). Peirce viveu entre 1839 e 1914 e não publicou artigos organizados sobre o tema. O que nos chega aos dias de hoje são fragmentos de textos que refletem o pensamento do filósofo sobre o assunto, colecionados após sua morte para efeito de publicação, principalmente nos anos da década de 1930 (Peirce, 1958). Lendo-os, observam-se mudanças conceituais e até mesmo de nomenclatura; por exemplo, abdução ora é denominada como tal, ora é tratada como presunção, retrodução e hipótese. Sendo assim, os conceitos de dedução, indução e abdução descritos nesta seção são produtos de interpretação da obra de Peirce e de algum consenso extraído junto a obras de pesquisadores contemporâneos (Aliseda, 2006; Magnani, 2009).

Raciocínios frequentemente são formalizados utilizando um sistema ternário que inclui um conjunto teoria, uma hipótese ou um conjunto de hipóteses e uma conclusão ou um conjunto de conclusões. Os nomes e papéis atribuídos aos elementos de uma estrutura variam de modo a melhor acomodar a um ou a outro discurso filosófico, teoria lógica empregada ou aplicação no âmbito da Inteligência Artificial. Como exemplos, (1) o silogismo regular clássico de origem grega, engloba três elementos conhecidos como premissa maior, premissa menor e conclusão; (2) em Lógica Proposicional, um teorema é definido como um conjunto de premissas que tem como consequência lógica uma conclusão.

Neste trabalho utilizaremos a nomenclatura mais comumente usada em trabalhos sobre abdução, na qual um raciocínio é composto por um conjunto teoria (T), um conjunto de hipóteses (H) e um conjunto de fatos (F), que podem ser entendidos, respectivamente, como premissa maior, premissa menor e conclusão.

No raciocínio dedutivo, parte-se dos conjuntos teoria e hipóteses para concluir o conjunto fatos que, em uma estrutura de raciocínio com foco explícito em dedução, frequentemente é denominada de conclusão. O Exemplo 1.1 descreve uma situação e a realização de um raciocínio dedutivo sobre a situação descrita.

**Exemplo 1.1.**<sup>1</sup> Considerando como exemplo a descrição de um circuito elétrico simples composto de uma lâmpada, uma bateria e um interruptor conectados por fios, a Figura 1.1 apresenta um raciocínio dedutivo: tendo por teoria que *se acionamos o interruptor então a lâmpada acende* e por hipótese que *acionamos o interruptor*, pode-se deduzir que *a lâmpada acende*. Isto é, assumindo uma interpretação na qual T e H são simultaneamente verdade então o raciocínio dedutivo garante a verdade de F.

**Conhecendo-se os conjuntos:**

T = {Se acionamos o interruptor então a lâmpada acende},

H = {Acionamos o interruptor},

---

**Pode-se deduzir o conjunto:**

F = {A lâmpada acende}.

**Figura 1.1: Um exemplo de raciocínio dedutivo.**

□

Nesta dissertação utilizaremos o símbolo □ para representar o fim de um exemplo ou de uma definição.

No raciocínio indutivo, parte-se de uma enumeração de correspondências entre hipóteses e fatos para induzir uma teoria. O Exemplo 1.2 apresenta um raciocínio indutivo. A enumeração de hipóteses e fatos apresentados no Exemplo 1.2 permite raciocinar para generalizar uma teoria. Cabe observar, entretanto, que interpretar as enumerações de hipóteses e fatos como verdade não garante que a teoria obtida seja verdade. Tudo o que se tem em geral, neste tipo de raciocínio, é uma boa razão para acreditar ou não que a teoria obtida seja verdade. Essa boa razão está associada à significância ou não da quantidade de enumerações (amostras) disponíveis em relação à população em que estão inseridas.

**Exemplo 1.2.** Novamente considerando como exemplo o funcionamento de circuitos elétricos simples, considere desta vez que estamos diante de vários interruptores e que, de alguma maneira, podemos confirmar que se ao acionarmos o interruptor 1 então a lâmpada 1 acenderá, se acionarmos o interruptor 2 a lâmpada 2 acende, se acionarmos o interruptor 3 a lâmpada 3 acende e assim, sucessivamente. A Figura 1.2 instancia esta situação nos conjuntos H, F e observações que correlacionam elementos de H a

---

<sup>1</sup> Posteriormente, raciocínios serão formalizados nesta dissertação por meio de Lógica Proposicional. Nestas seções introdutórias preferimos uma notação mais casual, com textos em português, pela sua generalidade.

elementos F. O conjunto teoria T é obtido a partir de um raciocínio indutivo que generaliza as observações que correlacionam H à T.

**Conhecendo-se:**

H = { O interruptor 1 é acionado,  
O interruptor 2 é acionado,  
O interruptor 3 é acionado,  
... }.

F = { A lâmpada 1 acende,  
A lâmpada 2 acende,  
A lâmpada 3 acende,  
... }.

Observações que correlacionam hipóteses a fatos:

- se o interruptor 1 é acionado então a lâmpada 1 acende.
- se o interruptor 2 é acionado então a lâmpada 2 acende.
- se o interruptor 3 é acionado então a lâmpada 3 acende.
- ...

---

**Pode-se induzir o conjunto:**

T = {Se o interruptor  $i$  é acionado então a lâmpada  $i$  acende}.

**Figura 1.2: Um exemplo de raciocínio indutivo.**

□

No raciocínio abduutivo o ponto de partida são os fatos observados, i.e., manifestações, sintomas, sinais etc. Partindo-se de um conjunto de fatos abduz-se um conjunto de hipóteses, que explicam os fatos, tendo por base uma teoria. Com T e F do Exemplo 1.3 pode-se raciocinar para encontrar H. Cabe observar que interpretar uma teoria e os fatos como verdade não garante que as hipóteses formuladas sejam verdade. Tal como no raciocínio indutivo, em geral o que se tem neste tipo de raciocínio é uma boa razão ou não para acreditar que certas hipóteses sejam verdade. Essa boa razão está associada a fatores que serão explorados no Capítulo 3, tais como a força explanatória e a simplicidade das hipóteses formuladas.

**Exemplo 1.3.** Ainda considerando o funcionamento de circuitos elétricos simples, composto de uma lâmpada, uma bateria e um interruptor, a Figura 1.3 apresenta como é instanciado um raciocínio abduutivo. A partir do conjunto teoria T e do conjunto de fatos F, o raciocínio abduutivo produz o conjunto de hipóteses H.

**Conhecendo-se os conjuntos:**

Teoria  $T = \{\text{Se acionamos o interruptor então a lâmpada acende}\}$ ,

Fatos  $F = \{\text{A lâmpada acende}\}$ ,

---

**Pode-se abduzir o conjunto:**

Hipóteses  $H = \{\text{Acionamos o interruptor}\}$ .

**Figura 1.3: Um exemplo de raciocínio abduzitivo.**

□

Finalizamos esta seção com um outro exemplo, adaptado de Peirce (1867), sobre como operam os raciocínios dedutivo, indutivo e abduzitivo (Tabela 1.1). O cenário deste exemplo inclui um saco de feijões e uma porção de feijão. Cada uma das três colunas da tabela apresenta um tipo de raciocínio. As duas primeiras linhas da tabela descrevem o que é conhecido, como ponto de partida do raciocínio, em cada tipo de raciocínio. A última linha da tabela apresenta o resultado de cada raciocínio.

**Tabela 1.1: Outro exemplo da diferença entre os raciocínios dedutivo, indutivo e abduzitivo (Adaptado de Peirce (1867)).**

	<i>Dedução</i>	<i>Indução</i>	<i>Abdução</i>
	$T = \{\text{Todos os feijões deste saco são brancos}\}$	$H = \{\text{Os feijões desta porção foram aleatoriamente selecionados deste saco}\}$	$T = \{\text{Todos os feijões deste saco são brancos}\}$
<i>Conhecendo-se</i>	$H = \{\text{Esta porção de feijão é deste saco}\}$	$F = \{\text{Os feijões desta porção são brancos}\}$	$F = \{\text{Os feijões desta porção são brancos}\}$
<i>Raciocina-se</i>	$F = \{\text{Os feijões desta porção são brancos}\}$	$T = \{\text{Todos os feijões deste saco são brancos}\}$ (possivelmente)	$H = \{\text{Esta porção de feijões é deste saco}\}$ (possivelmente)

## 1.2 – Abdução e indução

Para Peirce (1867) dedução, indução e abdução formam em conjunto um modelo triádico natural para descrever todo tipo de raciocínio.

Entretanto, a literatura apresenta-se de maneira muito diversificada sobre quais são e o que são os diferentes tipos de raciocínio. A influência da classificação de Mill (1862) é muito forte ainda hoje em dia. Para Mill existem os raciocínios dedutivos e os indutivos, esse último termo sendo usado para descrever todos os tipos de raciocínios não-dedutivos. Harman (1965) entende a abdução como a forma básica de raciocínio não-dedutivo e inclui a indução (enumerativa) como um tipo de abdução. Diferentemente, Heit (2007) emprega o termo indução para significar qualquer tipo de raciocínio que introduz incerteza e, sendo que a formulação de hipóteses abduativas possui incerteza, por conta da boa razão ou não para acreditar que certas hipóteses sejam verdade, caracteriza a abdução como um tipo de indução.

Há muita variação também na terminologia empregada para descrever raciocínios não dedutivos, seja em um sentido amplo ou estrito. No lugar de “abdução” é comum encontrar-se referências a “indução explanatória” ou “raciocínio explanatório” e, no lugar de “indução” frequentemente encontra-se “indução enumerativa”, “generalização indutiva” e “projeção indutiva”. O chamado “raciocínio estatístico” (Hacking, 2001) é um tipo de raciocínio não-dedutivo com fundamento em probabilidade que exhibe a mesma diversidade de fenômenos comumente atribuídos à abdução e indução.

Devido à existência destas diferentes concepções, este trabalho define o seu entendimento sobre estes termos da seguinte maneira. Abdução refere-se ao raciocínio que formula hipóteses para explicar fatos (evidências, manifestações, sinais etc.) tendo como fundamento uma teoria. Indução está restrita ao que se convencionou chamar de ‘indução enumerativa’, raciocínio que parte de amostras para generalizar sobre o todo. A Figura 1.4 apresenta dois exemplos de indução, segundo o sentido estrito considerado neste texto. A partir das enumerações (amostras), raciocina-se para desenvolver teorias gerais. Enquanto que indução não necessita de uma teoria intrínseca, abdução depende de uma teoria de fundo para que hipóteses sejam formuladas e testadas (Aliseda, 2006).

**Conhecendo-se que:**

- Cão é mamífero e tem coração,
- Gato é mamífero e tem coração,
- Rato é mamífero e tem coração,
- ....

---

**Pode-se raciocinar que:**

Todo mamífero tem coração.

(a)

**Conhecendo-se que:**

- $1 = 1$ ,
- $1 + 3 = 4$ ,
- $1 + 3 + 5 = 9$ ,
- ....

---

**Pode-se raciocinar que:** $1 + 3 + 5 + \dots + 2n - 1 = n^2$  ( $n \geq 1$ , inteiro).

(b)

**Figura 1.4: Exemplos de raciocínio indutivo segundo a concepção empregada neste trabalho. A partir de amostras de um todo, raciocina-se para estabelecer sentenças genéricas para o todo: (a) Em Biologia, “todo mamífero tem coração”; e (b) Em Matemática, “a soma dos  $n$  primeiros números ímpares é igual a  $n^2$ ”.**

Abdução e indução, entretanto têm também similaridades. Abdução e indução são raciocínios que se processam das evidências para as explicações, dos efeitos para as causas, ou ainda, usando termos da Lógica, partem das conclusões para as premissas, em uma direção oposta ao da dedução clássica. São raciocínios que, em geral, se dão a partir de um conhecimento parcial do mundo, i.e., conhecimento de apenas alguns fatos (abdução) e conhecimento de algumas amostras de uma população (indução). Por fim, ambos são tipos de raciocínio não-monotônicos, isto é, novas premissas podem invalidar um argumento considerado previamente válido. Por exemplo, uma generalização obtida por raciocínio indutivo, e.g., todos os mamíferos são animais endotérmicos, i.e., controlam a própria temperatura corporal, permanece válida até a adição de uma premissa, e.g., ratos-toupeira-pelado são mamíferos e não são endotérmicos, que invalida a generalização e as consequências prévias dela. Em abdução, uma teoria, e.g., Teoria da Geração Espontânea, segundo a qual vida pode surgir espontaneamente a partir de matéria viva ou morta, que explica, e.g., que larvas podem surgir a partir de carne em decomposição, pode ser refutada juntamente com suas consequências prévias quando fatos novos são obtidos, e.g., larvas não surgem em carnes em decomposição colocadas em frascos selados, e também, naqueles fechados com uma gaze.

### 1.3 – Exemplos de uso de raciocínio abduativo

Racocínios abduativos são comuns e permeiam intensamente as nossas vidas. Em situações corriqueiras do dia a dia, expressões tais como “Eu desconfio que ...”, “Hum? Como pode ser que ...”, e “Ah! Neste mato tem coelho.” sugerem que alguém fez um raciocínio abduativo. Esta seção, como motivação, apresenta uma variedade de cenários

que ilustram o emprego de raciocínio abduutivo. Estes cenários variam desde situações comuns do cotidiano até o uso de raciocínio abduutivo em ciência. Eles são úteis para explicitar a abrangência do uso de raciocínio abduutivo e também para as discussões dos demais capítulos deste trabalho. Mais situações podem ser encontradas no Apêndice I.

### **Cenário 1.1 – Situação do cotidiano**

José possui na frente de sua casa um grande gramado. Um dia José chega à sua casa e constata o fato de que o gramado está molhado. Sabendo-se que chuva pode molhar o gramado e que, também, irrigadores instalados podem molhar o gramado, pode-se formular as hipóteses de que:

- Choveu, ou
- Os irrigadores foram ligados, ou
- Choveu e os irrigadores estiveram ligados.

Este cenário é uma variação de um exemplo de raciocínio abduutivo que se tornou um clássico da literatura. Em relação ao enquadramento deste cenário nos conjuntos teoria (T), hipóteses (H) e fatos (F), pode-se dizer que T engloba os conhecimentos de que se chover ou os irrigadores forem ligados o gramado fica molhado:  $T = \{\text{Se chover ou os irrigadores forem ligados então o gramado fica molhado}\}$ . José observa um único fato para o conjunto F, que é o fato do gramado estar molhado:  $F = \{\text{O gramado está molhado}\}$ . Diante de T e F pode-se desenvolver hipóteses abdutivas tais como  $H = \{\text{Choveu, Os irrigadores foram ligados, Choveu e os irrigadores foram ligados}\}$ .

### **Cenário 1.2 – Situação do cotidiano**

Considere novamente José passando pela mesma situação descrita no Cenário 1. No entanto, José sabe que o tanque de abastecimento dos irrigadores está sem água e que sem água nos tanques os irrigadores não ligam. Agora, pode-se dizer que a única hipótese que explica o fato do gramado estar molhado é que:

- Choveu.

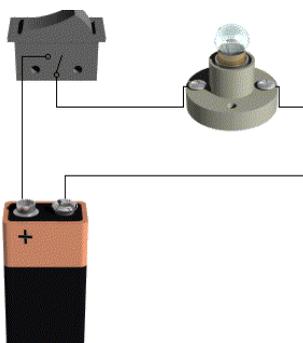
A novidade deste cenário, em relação ao Cenário 1.1, é que o raciocínio abduutivo para explicar o gramado molhado encontra-se com a seguinte condição para sua realização: os irrigadores não ligam sem que haja água no tanque que os abastecem e José sabe que não há água nos tanques. Neste caso a hipótese de que os irrigadores estão ligados é descartada, pois é conflitante com as condições para a realização do raciocínio abduutivo.

### Cenário 1.3 – Diagnóstico médico

Um médico observa que um paciente apresenta sintomas (fatos) de febre, dor de cabeça e coriza. Tendo conhecimento geral de Medicina (teoria), o médico cria um diagnóstico, isto é, raciocina para obter as possíveis doenças do paciente (hipóteses).

### Cenário 1.4 – Descoberta de falhas em sistemas com comportamento conhecido

A Figura 1.5 apresenta um circuito elétrico muito simples constituído por um interruptor, uma lâmpada conectada a um soquete, uma bateria e fios conectando estes componentes.



**Figura 1.5: Um circuito elétrico simples.**

Foram constatados os seguintes fatos:

- Se o interruptor é acionado então a lâmpada não acende.
- A bateria tem carga.
- A lâmpada não está “queimada”.

Considerando o funcionamento do circuito (teoria) e os fatos apresentados (fatos), pode-se levantar, entre outras, as hipóteses (hipóteses):

- De que o interruptor está com defeito e/ou
- Os fios estão rompidos e/ou
- Há problemas de conexão entre um fio e um dispositivo.

### Cenário 1.5 – Intencionalidade no raciocínio abduutivo – acusação e defesa

No Bairro 27 houve um crime. José foi morto por um ferimento de faca no coração. Pedro foi quem acionou a polícia dizendo ter encontrado José morto. Não há testemunhas oculares sobre o ocorrido. Após ser indiciado, Pedro é réu em um julgamento no qual é acusado de matar José. A única prova material levada ao tribunal são as digitais de Pedro na faca que matou José.

Neste cenário de julgamento, a lei (teoria) e as provas (fatos) estão à disposição e são as mesmas tanto para advogado de defesa quanto para o advogado de acusação. O que os diferencia são suas conhecidas intenções. O advogado de acusação tem a intenção de incriminar o réu, enquanto que o advogado de defesa tem a intenção de defendê-lo.

Baseados em distintas intenções, cada um deles formula distintas hipóteses para explicar o que aconteceu entre José e Pedro. O advogado de acusação pode formular a hipótese de que Pedro matou José, e isto explica por que as impressões digitais de Pedro estão na faca utilizada como prova. Em contrapartida, o advogado de defesa pode formular a hipótese de que Pedro tentou retirar a faca do coração de José para socorrê-lo, por isto as suas impressões digitais estão nela.

Este cenário é interessante porque explicita mais condições no raciocínio abduativo. Raciocínios abduativos podem ser essencialmente intencionais, isto é, são afetados pela intenção daquele que raciocina e, por mais estranho que isto possa parecer para aqueles acostumados com raciocínios dedutivos, o resultado de um raciocínio abduativo pode depender de condições.

### **Cenário 1.6 – Descoberta científica**

Este cenário é baseado em um exemplo descrito em um estudo de Thagard (1989). Neste estudo sobre coerência explanatória de teorias Thagard contrapõe diversas teorias científicas, entre elas a Teoria de Lavoisier para a Combustão e Calcinação (fenômeno que hoje é conhecido como oxidação) e a obsoleta Teoria do Flogisto.

Formulada em 1667 por Johann Becker, a Teoria do Flogisto estabelece que certos corpos contêm uma substância chamada flogisto. Segundo esta teoria:

- Corpos combustíveis contêm flogisto.
- Na combustão, o flogisto é despreendido.
- O flogisto pode passar de um corpo para o outro.
- Metais contêm flogisto.
- Na calcinação, o flogisto é despreendido.

Na década de 1760 Lavoisier observou em laboratório os seguintes fatos:

- A combustão ocorre somente na presença de ar.
- O incremento de massa de um corpo queimado é exatamente igual à perda de peso do ar no qual ele é queimado.
- Os metais estão sujeitos à calcinação.
- Na calcinação os corpos incrementam sua massa.
- Na calcinação o volume do ar é diminuído.
- Na combustão calor e luz são emitidos e a inflamação é transmitida de um corpo para outro.

Os novos fatos observados por Lavoisier (aumento de massa de corpos queimados e metais calcinados) não podem ser explicados pela Teoria do Flogisto (o flogisto deveria ter massa negativa). Então Lavoisier desenvolve uma nova teoria segundo a qual existe um elemento no ar que participa da combustão de corpos e da calcinação de metais.

Em todos os cenários anteriores, hipóteses foram formuladas para explicar fatos tendo em vista uma teoria e, em alguns casos, certas condições. Este cenário contém algo novo. Uma teoria é incapaz de ser utilizada para formular hipóteses para explicar os novos fatos que estão sendo descobertos e, então, uma nova teoria está sendo desenvolvida. Em geral, durante um único raciocínio abduutivo a teoria é admitida como verdade inquestionável. Entretanto, a impossibilidade de gerar hipóteses consistentes com uma teoria sugere que ela precisa ser reformulada ou substituída por outra.

#### **1.4 – Critérios para a seleção de boas hipóteses**

De forma geral, podem existir várias hipóteses para explicar fatos observados. Entretanto, algumas hipóteses podem explicar melhor os fatos do que outras. Assim, raciocínios abdutivos podem ser vistos como processos de formulação de  $m \geq 0$  hipóteses gerais seguido da seleção de  $n \leq m$  boas hipóteses. Evidentemente, é necessário o estabelecimento de critérios para tal seleção, mas o que caracteriza uma hipótese boa não tem uma resposta fácil.

Filósofos contemporâneos têm analisado a questão da seleção de boas hipóteses. Harman (1965) entende abdução como inferência para a melhor explicação e argumenta que a melhor hipótese é a mais simples, a mais plausível, a que explica mais os fatos observados, e é menos *ad hoc*, i.e., criada especificamente para explicar o que se quer explicar.

Thagard (1978) busca na contraposição entre teorias, e.g., Teoria da Evolução de Darwin *versus* Teoria Criacionista; Teoria da Combustão de Lavoisier *versus* Teoria do Flogisto, critérios que explicam a seleção de uma teoria (ou hipótese) em relação a outra. Thagard estabelece que a melhor hipótese é a mais consiliente (explica mais fatos), a mais simples e a que tem maior analogia com hipóteses que explicam outros fatos. Estes conceitos são trabalhados formalmente para criar em Thagard (1989) um algoritmo conexionista que analisa a coerência explanatória de uma teoria de acordo com os critérios de consiliência, simplicidade e analogia.

Em Dillig & Dillig (2013), encontramos um critério denominado *generalidade*. Hipóteses com maior generalidade são hipóteses que explicam tudo o que outras hipóteses podem explicar e, possivelmente, explicam mais fatos. Por exemplo,

considerando  $\alpha$  e  $\beta$  como proposições e considerando que  $\alpha \models \beta$ , e  $\beta \not\models \alpha$ , diz-se que  $\alpha$  possui mais generalidade que  $\beta$ , pois  $\alpha$  pode explicar tudo o que  $\beta$  explica e, possivelmente, explica também mais fatos.

Crítérios para seleção de boas hipóteses têm sido extensivamente considerados em outros trabalhos de Filosofia, Psicologia e Inteligência Artificial, mas uma formulação precisa deles permanece controversa. De forma geral, os critérios de *força explanatória* e *simplicidade* de uma hipótese são recorrentes e têm conotações semelhantes em vários trabalhos. Por isto, este trabalho propõe, para a seleção de boas hipóteses, o emprego destes dois critérios.

Tendo como objetivo o desenvolvimento de um algoritmo para abdução, que necessita lidar com medidas quantitativas para estes critérios, este trabalho propõe o entendimento destes critérios como descrito a seguir:

- *Força explanatória*: apesar de ser abordada com diferentes nomes e alguma diferença de conotação – mais explicativa (Harman, 1965), consiliência (Thagard, 1978), relevância (Magnani, 2009) – a força explanatória de uma hipótese refere-se ao quanto ela é abrangente e unificadora. Uma medida para a força explanatória de uma hipótese é a quantidade de fatos que ela é capaz de explicar. Assim, por exemplo, uma hipótese que explica 4 de 5 fatos observados, deve ser preferida em relação a outra que explica apenas 2 fatos.
- *Simplicidade*: este trabalho assume a tentativa de Aliseda (2006) e Dillig & Dillig (2013) para a captura do conceito de simplicidade segundo a qual, deve-se preferir, por exemplo, uma hipótese contendo uma explicação, e.g., “ $a$ ”, em vez de uma hipótese contendo mais de uma explicação, e.g., “ $a$  e  $b$ ”, ou preferir “ $a$  e  $b$ ” em vez de “ $a$ ,  $b$  e  $c$ ”.

Assim, como será detalhadamente discutido no Capítulo 3, em um sistema de raciocínio que admite um conjunto de  $m \geq 0$  de hipóteses que explicam parcialmente ou totalmente um conjunto não vazio de fatos, a seleção de  $n \leq m$  boas hipóteses será pautada pela seleção das hipóteses com maiores forças explanatórias e, dentre elas, as mais simples.

## 1.5 – Organização da Dissertação

Este trabalho propõe um algoritmo polinomial para abdução Peirceana baseado em uma estrutura de raciocínio quaternária usando Lógica Proposicional.

O Capítulo 2 aborda conceitos da Lógica Proposicional utilizados nesta dissertação. Ele também descreve a formalização para raciocínios abduativos em Lógica. Em seguida apresenta abordagens existentes para a realização de raciocínios abduativos em Lógica bem como suas desvantagens. O Capítulo 2 também declara os objetivos do trabalho e a metodologia utilizada para atingi-los.

De forma geral, os algoritmos existentes operam sobre uma estrutura de raciocínio ternária que denominamos THF (Teoria, Hipóteses e Fatos). Este trabalho propõe a extensão da estrutura THF para a estrutura quaternária TCHF pela adição de um conjunto C, de condições aceitas na realização de raciocínios abduativos. Algoritmos para abdução frequentemente utilizam algum critério para a escolha de boas hipóteses. Este trabalho propõe um critério simples, fácil de implementar, e que tem se mostrado, nos experimentos que realizamos, eficaz para eleger boas hipóteses entre várias hipóteses candidatas. O Capítulo 3 apresenta a estrutura TCHF e o critério proposto para a seleção de boas hipóteses.

O Capítulo 4 descreve o algoritmo, denominado Peirce, que é a solução deste trabalho para abdução. São apresentados também neste capítulo os resultados de uma pesquisa experimental que teve por objetivo verificar a coincidência entre os raciocínios abduativos realizados por pessoas e pelo algoritmo proposto.

Uma implementação do algoritmo Peirce é descrita no Capítulo 5. Trata-se de um sistema baseado na Web que permite o desenvolvimento de modelos e a realização de raciocínios abduativos sobre os modelos criados.

O Capítulo 6 tenta lançar uma visão crítica sobre este trabalho na forma de discussões e algumas conclusões.

Uma série de apêndices complementam a dissertação oferecendo uma referência adicional ao texto principal. De especial interesse para avaliação deste trabalho de conclusão de mestrado é o Apêndice V que reproduz um artigo que publicou uma parte do conteúdo desta dissertação.

# Capítulo 2

## Contexto de Estudo, Objetivo e Metodologia

*A Seção 2.1 aborda conceitos da Lógica Proposicional, utilizados nesta dissertação. A Seção 2.2 apresenta uma estrutura para a formalização de raciocínio abduativos denominada THF. A Seção 2.3 apresenta abordagens existentes para a realização de raciocínios abduativos em Lógica. A Seção 2.4 apresenta as desvantagens das abordagens existentes. A Seção 2.5 remete-se ao objetivo deste trabalho em relação a realização do raciocínio abduativo e a tentativa de minimizar as desvantagens identificadas. A Seção 2.6 apresenta a metodologia empregada na realização do trabalho e dos experimentos que o compõem.*

## 2.1 – Lógica Proposicional

A Lógica Proposicional (LP) é um formalismo lógico que pode ser utilizado para a formalização de formas de raciocínio. Nesta seção apresentamos conceitos da LP que são utilizados nesta dissertação. Esta seção tem como intuito situar o leitor ao foco deste trabalho que é o raciocínio abduutivo formalizado em lógica.

A gramática da linguagem da LP e as notações utilizadas nesta seção estão adaptadas para o uso que faremos delas nesta dissertação. Howard (1999) e Nicoletti (2010) podem ser utilizados como complemento ao que aqui é apresentado em relação à Lógica.

Iniciamos esta seção apresentando a linguagem da LP em seus aspectos sintáticos para posteriormente abordar seus aspectos semânticos. O formalismo EBNF (Definição 2.1) será empregado para descrever a linguagem da LP.

**Definição 2.1 (Extended Backus-Naur Form):** A EBNF é um formalismo para descrever gramáticas que utilizam a seguinte notação:

- $\langle a \rangle$  representa o símbolo não-terminal (nome) “a”;
- Termos em **negrito** representam símbolos terminais;
- $\{ \alpha \}$  indica a repetição da cadeia de símbolos “ $\alpha$ ” zero ou mais vezes;
- $[ \alpha ]$  indica que a cadeia de símbolos “ $\alpha$ ” é opcional.
- $\alpha \mid \beta$  indica uma escolha entre as cadeias de símbolos “ $\alpha$ ” e “ $\beta$ ”.
- $::=$  é o símbolo usado para indicar uma produção gramatical.

□

A linguagem da LP é expressa por proposições (Definição 2.2).

**Definição 2.2 (Proposição):** Uma proposição pertence à linguagem da LP se for gerada pela seguinte gramática, descrita pelo formalismo EBNF.

```
<proposição> ::=
¬<proposição> |
<proposição> Λ <proposição> |
<proposição> V <proposição> |
<proposição> → <proposição> |
```

```

<proposição> ↔ <proposição> |
(<proposição>) |
<átomo>
<átomo> ::= <letra> | { <letra> | <dígito> | <subtraço> }
<letra> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n
| o | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D |
E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U
| V | W | X | Y | Z
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<subtraço> ::= _

```

□

**Exemplo 2.1.** São exemplos de proposições:  $p$ ,  $q$ ,  $r$ ,  $p \wedge q$ , *chuva*,  $chuva \vee sol$ ,  $chuva \rightarrow gramado$ ,  $Molhado$ ,  $(a \rightarrow b) \wedge c$ .

□

Os símbolos  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftrightarrow$  são denominados conectivos lógicos e são classificados de acordo com a Definição 2.3.

**Definição 2.3 (Classificação dos conectivos lógicos):** Os conectivos lógicos  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftrightarrow$  são classificados da seguinte maneira:

- $\neg$ : não / negação;
- $\wedge$ : e / conjunção;
- $\vee$ : ou / disjunção;
- $\rightarrow$ : se... então / implicação;
- $\leftrightarrow$ : se e somente se / dupla implicação.

□

A utilização de conectivos lógicos permite a expressão de proposições compostas como, por exemplo,  $p \wedge q \vee r$ , e com significados semânticos que serão mais tarde tratados nesta mesma seção. A gramática da Definição 2.2 é incapaz de definir sem ambiguidade sintática proposições compostas, e.g.,  $p \wedge q \vee r$ , que pode ser lida (1)  $p \wedge q$  primeiro e, posteriormente a proposição composta resultante de  $p \wedge q$  com a adição de  $\vee r$  ou (2)  $q \vee$

$r$  primeiro e, posteriormente a proposição composta resultante de  $q \vee r$  com a adição de  $p \wedge$ . A Definição 2.4, determina a ordem de leitura de proposições da linguagem da LP.

**Definição 2.4 (Ordem de leitura de proposições):** A ordem de leitura de proposições obedece às seguintes regras:

1. Proposições entre parêntesis devem ser lidas primeiro.
2. A leitura de proposições compostas por conectivos lógicos obedece à ordem de prioridade:  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftrightarrow$ , a qual o conectivo mais à esquerda tem maior prioridade.
3. A leitura de proposições compostas por conectivos lógicos idênticos deve ser feita da esquerda para a direita.

□

**Exemplo 2.2.** A proposição  $p \vee q \wedge r$  deve ser lida como  $p \vee (q \wedge r)$ , pois o conectivo  $\wedge$  possui maior prioridade do que o conectivo  $\vee$ . A proposição  $p \rightarrow q \rightarrow r$  deve ser lida como  $(p \rightarrow q) \rightarrow r$ , pois, sendo formulada com conectivos que possuem a mesma prioridade, deve ser lida da esquerda para a direita.

□

**Definição 2.5 (Tipos de proposições):** Proposições são classificadas das seguintes formas:

- Proposição atômica ou átomo, quando é expressa por apenas um átomo;
- Literal, quando é expressa por um átomo ou átomo negado (átomo precedido pelo conectivo lógico  $\neg$ );
- Proposição composta, quando é expressa por dois ou mais literais conectados por conectivos lógicos.

□

Na linguagem da LP proposições possuem um aspecto sintático, definido pelos símbolos que as expressam, e um aspecto semântico, referente a seu valor verdade. A semântica de uma proposição é descrita na Definição 2.6.

**Definição 2.6 (Semântica de uma proposição):** Semanticamente uma proposição pode assumir valores verdade: Verdade (V) ou Falso (F).

□

Uma interpretação (Definição 2.7) é a associação entre a sintaxe e a semântica de uma proposição.

**Definição 2.7 (Interpretação de uma proposição):** A interpretação de uma proposição refere-se ao valor verdade associado à uma proposição. A interpretação de uma proposição pode ser V ou F. Se dizemos que a proposição *choveu* é verdade então a interpretação de *choveu* é V, contrariamente se dizemos que *choveu* é falso então a interpretação de *choveu* é F.

□

**Exemplo 2.3.** Exemplos de interpretação de proposições.

- $interruptorLigado = V$ ;
- $lampadaAcesa = F$ ;
- $Choveu \wedge Molhou = V$ ;
- $pneumonia \rightarrow tosse = V$ .

□

Ao compor proposições, utilizando os conectivos lógicos, estas proposições assumem novas interpretações. Essas interpretações variam de acordo com as interpretações das proposições que às compõem e a semântica dos conectivos lógicos. A Definição 2.8 descreve a semântica dos conectivos lógicos.

**Definição 2.8 (Semântica dos conectivos lógicos):** A semântica dos conectivos lógicos, que determinam a interpretação de proposições compostas, obedece as seguintes regras.

- $\neg$ : Inverte a semântica de uma proposição;
- $\wedge$ : A interpretação da proposição contendo este conectivo é V, somente se as proposições conectadas pelo conetivo possuírem interpretação V;
- $\vee$ : A interpretação da proposição contendo este conectivo é V, se pelo menos uma das proposições conectadas pelo conetivo possuírem interpretação V;
- $\rightarrow$ : A interpretação da proposição contendo este conectivo é V, se ambas as proposições conectadas pelo conectivo possuírem interpretação V ou se a proposição à esquerda do conetivo possuir interpretação F;

- $\leftrightarrow$ : A interpretação da proposição contendo este conectivo é V, somente se ambas as proposições conectadas pelo conectivo possuírem as mesmas interpretações.

□

**Exemplo 2.4.** Como exemplo de interpretações de proposições utilizaremos as proposições  $p$  e  $q$ . A Tabela 2.1, também conhecida como Tabela Verdade, apresenta todas as combinações possíveis de interpretações de  $p$  e  $q$  e suas interpretações utilizando os conectivos lógicos, sendo cada linha uma possível interpretação referente aos valores verdade de  $p$  e  $q$ .

**Tabela 2.1: Tabela Verdade.**

$p$	$q$	$\neg p$	$\neg q$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
V	V	F	F	V	V	V	V
V	F	F	V	F	V	F	F
F	V	V	F	F	V	V	F
F	F	V	V	F	F	V	V

□

Semelhante à leitura sintática, as interpretações obedecem à uma ordem de interpretação (Definição 2.9).

**Definição 2.9 (Ordem de interpretação de proposições):** A ordem de interpretação de proposições segue as mesmas regras da ordem de leitura das proposições definidas na Definição 2.4.

□

**Exemplo 2.5.** Considerando a proposição  $p \rightarrow q \rightarrow r$  a interpretação desta proposição se dá primeiro interpretando  $p \rightarrow q$  e, posteriormente, armazenando as interpretações resultantes em uma proposição  $\alpha$ , interpreta-se  $\alpha \rightarrow r$ .

**Tabela 2.2:** Ordem de interpretação de proposições, neste exemplo primeiro interpreta-se  $p \rightarrow q$  armazenando as interpretações de  $p \rightarrow q$  em  $\alpha$ , para posteriormente interpretar  $\alpha \rightarrow r$ , assim seguindo a ordem de interpretação de proposições, neste exemplo conectivos de mesma prioridade.

$p$	$q$	$r$	$p \rightarrow q$	$\alpha \rightarrow r$
V	V	V	V	V
V	V	F	V	F
V	F	V	F	V
V	F	F	F	V
F	V	V	V	V
F	V	F	V	F
F	F	V	V	V
F	F	F	V	F

□

De acordo com as interpretações de uma proposição, pode-se classifica-la como uma contradição (Definição 2.10).

**Definição 2.10 (Contradição):** Uma proposição é uma contradição ( $\perp$ ) se todas as suas possíveis interpretações são F.

□

**Exemplo 2.5.** A proposição  $p \wedge \neg p$  é uma contradição, pois tem todas as possíveis interpretações F, de acordo com a Tabela 2.3.

**Tabela 2.3: Exemplo de contradição proposição  $p \wedge \neg p$ .**

$p$	$\neg p$	$p \wedge \neg p$
V	F	F
F	V	F

□

A Definição 2.11 define uma consequência lógica.

**Definição 2.11** ( $\alpha \models \beta$ , i.e.,  $\beta$  é consequência lógica de  $\alpha$ ). Sejam  $\alpha$  e  $\beta$  proposições. Então  $\alpha \models \beta$ , se e somente se, em qualquer interpretação em que  $\alpha$  é V, a interpretação de  $\beta$  também é V.

□

**Exemplo 2.6.** Tomemos como exemplo a proposição  $\alpha = (p \rightarrow q) \wedge p$  e a proposição  $\beta = q$ , a Tabela 2.4 apresenta as interpretações de  $\alpha$  e  $\beta$ .

**Tabela 2.4:** Um exemplo de consequência lógica, no caso a proposição  $\beta = q$  é consequência lógica da proposição  $\alpha = (p \rightarrow q) \wedge p$ , ou seja,  $(p \rightarrow q) \wedge p \models q$ .

$p$	$q$	$p \rightarrow q$	$(p \rightarrow q) \wedge p$
V	V	V	V
V	F	F	F
F	V	V	F
F	F	V	F

Nota-se que em todas as interpretações em que a proposição  $(p \rightarrow q) \wedge p$  é interpretada V, a proposição  $q$  também é interpretada V. Isto caracteriza que  $\alpha \models \beta$ , i.e., a proposição  $q$  é consequência lógica da proposição  $(p \rightarrow q) \wedge p$ .

□

Proposições com interpretações iguais podem ser escritas de diversas maneiras, e.g., a proposição  $\alpha = p \rightarrow q$ , pode ser escrita como a proposição  $\beta = \neg p \vee q$ .

**Tabela 2.5:** Exemplo de proposições diferentes com a mesma interpretação, no caso as interpretações da proposição  $\alpha = p \rightarrow q$  são iguais às interpretações da proposição  $\beta = \neg p \vee q$ .

$p$	$q$	$p \rightarrow q$	$\neg p \vee q$
V	V	V	V
V	F	F	F
F	V	V	V
F	F	V	V

Esta característica é denominada Equivalência Lógica e é descrita pela Definição 2.12.

**Definição 2.12 ( $\alpha \equiv \beta$ , i.e.,  $\alpha$  é equivalente a  $\beta$ ):** Sejam  $\alpha$  e  $\beta$  proposições. Então  $\alpha \equiv \beta$ , se e somente se,  $\beta$  for consequência lógica de  $\alpha$  e  $\alpha$  for consequência lógica de  $\beta$ , i.e.,  $\alpha \vDash \beta$  e  $\beta \vDash \alpha$ .

□

A Tabela 2.6 reúne algumas equivalências lógicas importantes para o contexto desta dissertação.

**Tabela 2.6: Equivalências Lógicas.**

<i>Equivalência</i>	<i>Nome</i>
$p \rightarrow q \equiv \neg p \vee q$	Equivalência lógica da condicional
$\neg(\neg p) \equiv p$	Lei da dupla negação
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Leis associativas
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Leis distributivas
$\neg(p \wedge q) \equiv (\neg p \vee \neg q)$ $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$	Leis De Morgan

As vantagens de uma padronização da forma com que proposições são descritas incluem a facilidade tanto de identificar quanto de comparar proposições. Uma padronização comum de escrita de proposições é denominada Forma Normal Conjuntiva (FNC).

Proposições na FNC (Definição 2.15) são compostas de uma conjunção de proposições disjuntivas. Proposições disjuntivas são denominadas cláusulas, (Definição 2.13).

**Definição 2.13 (Cláusula):** Uma cláusula é uma disjunção de literais, i.e.,  $l_1 \vee l_2 \vee l_3 \dots l_n$ , e cada  $l_i$  ( $1 \leq i \leq n$ ) é um literal.

□

**Exemplo 2.7.** São exemplos de cláusulas:  $p$ ,  $p \vee q$ ,  $p \vee \neg q \vee r$ .

□

**Definição 2.14 (Cláusulas Horn):** Uma cláusula é dita ser Horn se for expressa por, no máximo, um átomo.

□

**Exemplo 2.7.** São exemplos de cláusulas Horn:  $p$ ,  $p \vee \neg q$ ,  $\neg p \vee \neg q \vee \neg r$ .

□

**Definição 2.15 (Forma Normal Conjuntiva):** Uma proposição encontra-se na FNC, se e somente se, for uma conjunção de cláusulas, i.e.,  $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \dots \wedge \alpha_n$  em que cada  $\alpha_i$  ( $1 \leq i \leq n$ ) é uma cláusula. Se as cláusulas da FNC forem cláusulas Horn então a FNC é dita ser uma FNC com cláusulas Horn.

□

**Exemplo 2.8.** São exemplos de proposições na FNC:  $p$ ,  $p \vee q$ ,  $(p \vee q) \wedge r$ ,  $(p \vee q) \wedge (r \vee s)$ . São exemplos de proposições na FNC com cláusulas Horn:  $p$ ,  $p \vee \neg q$ ,  $(p \vee \neg q) \wedge \neg r$ ,  $(\neg p \vee \neg q) \wedge (\neg r \vee \neg s)$ .

□

Particularmente, este trabalho usa uma restrição da linguagem da LP denominada Forma HF (Definição 2.16). A Forma HF restringe a escrita de proposições da LP para um formato que criará a possibilidade de conversão destas proposições para proposições na FNC com cláusulas Horn.

**Definição 2.16 (Forma HF).** Uma proposição pertencente a linguagem da LP na Forma HF é uma proposição expressa em qualquer dos seguintes formatos:

- **Forma 1:**  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são literais.
- **Forma 2:**  $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são átomos negados.
- **Forma 3:**  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n \rightarrow b_1 \wedge b_2 \wedge b_3 \wedge \dots \wedge b_m$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são átomos e  $b_j$  ( $1 \leq j \leq m$ ) são literais.
- **Forma 4:**  $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n \rightarrow b_1 \vee b_2 \vee b_3 \vee \dots \vee b_m$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são literais e  $b_j$  ( $1 \leq j \leq m$ ) são átomos negados.

□

**Exemplo 2.9.** São exemplos de proposições na Forma HF:

- **Forma 1:**  $p$ ,  $\neg p$ ,  $p \wedge q$ ,  $\neg p \wedge \neg q \wedge r$ ;
- **Forma 2:**  $\neg p$ ,  $\neg p \vee \neg q$ ,  $\neg p \vee \neg q \vee \neg r$ ;

- **Forma 3:**  $p \rightarrow q, p \wedge q \rightarrow r, p \rightarrow \neg q \wedge \neg r \wedge s$ ;
- **Forma 4:**  $p \rightarrow \neg q, p \vee \neg q \rightarrow \neg r, \neg p \vee q \rightarrow \neg r \vee \neg s$ .

□

Podemos caracterizar o problema de determinar se uma proposição  $\beta$  é consequência lógica de uma proposição  $\alpha$  como a Prova de um Teorema. Provas de Teoremas em LP podem ser feitas seguindo diversos mecanismos. Um destes mecanismos é conhecido como Resolução. Este é um mecanismo conveniente para algoritmos que empregam Provas de Teoremas porque é baseado em apenas uma regra de inferência, conhecida como Regra de Resolução, descrita na Definição 2.17.

**Definição 2.17 (Regra de Resolução):** Sejam  $\alpha$  e  $\beta$  duas cláusulas. Se em  $\alpha$  encontra-se um átomo, e.g.,  $p$  e em  $\beta$  encontra-se o mesmo átomo, porém negado, e.g.,  $\neg p$ , o resolvente de  $(\alpha, \beta, p)$  consiste em: (1) retirar o átomo  $p$  de  $\alpha$ ; (2) retirar o átomo negado  $\neg p$  de  $\beta$  e; (3) unir as cláusulas resultantes. Se uma cláusula resolvente for vazia ela é dita cláusula vazia e sua interpretação é F.

□

**Exemplo 2.10.** Se aplicarmos a Regra de Resolução sobre as cláusulas  $\alpha = \neg p \vee q$ , e  $\beta = p \vee s$ , em relação ao átomo  $p$ , a cláusula resolvente  $(\alpha, \beta, p)$  que obtemos é a cláusula  $q \vee s$ .

□

Uma estratégia para Prova de Teoremas é a Prova por Refutação, Supondo que  $\alpha \models \beta$  então  $\alpha \wedge \neg\beta$  deve ser uma contradição, i.e., se  $\alpha \wedge \neg\beta$  for uma contradição então  $\alpha \models \beta$ .

**Exemplo 2.11.** Considerando  $\alpha = (p \rightarrow q) \wedge p$  e  $\beta = q$ , vimos no Exemplo 2.6 que  $\alpha \models \beta$ , a Tabela 2.7 mostra que  $\alpha \wedge \neg\beta$  é uma contradição.

**Tabela 2.7: Exemplo de Prova por Refutação, para provar que  $\alpha \models \beta$  então  $\alpha \wedge \neg\beta$  deve ser uma contradição.**

$p$	$q$	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$(p \rightarrow q) \wedge p \wedge \neg q$
V	V	V	V	F
V	F	F	F	F
F	V	V	F	F
F	F	V	F	F

□

A Regra de Resolução juntamente com a estratégia de Prova por Refutação serão utilizadas para estabelecer um algoritmo que implementa o mecanismo de Resolução para provar consequências lógicas.

## 2.2 – A Lógica como formalismo para o raciocínio abduativo

Diferentes abordagens podem ser utilizadas para formalizar o raciocínio abduativo. Pode-se descrever o problema do raciocínio abduativo como um problema clássico de busca (Romdhane e Ayeb, 2011), como um problema de cobertura de conjuntos, ou um problema de raciocínio probabilístico sobre Redes Bayesianas (Hacking, 2001).

De especial interesse para este trabalho é a formalização por meio da Lógica Proposicional, que descreve o raciocínio abduativo como a Prova de um Teorema. Como veremos a seguir em lógica a realização de um raciocínio abduativo consiste em encontrar proposições, que representam hipóteses, que unidas às proposições que representam uma teoria tenham como consequência lógica as proposições que representam fatos.

Algoritmos para raciocínios abduativos (Kakas & Kowalski, 1995; Aliseda 2006; Christiansen, 2008; Dillig & Dillig, 2013;) baseados em lógica operam sobre uma estrutura ternária de raciocínio, aqui denominada THF, que é frequentemente instanciada de acordo com a Definição 2.19.

**Definição 2.19 (Estrutura THF):** A estrutura THF para formalização de raciocínios abduativos é um sistema  $\langle T, H, F \rangle$  consistindo em:

- Um conjunto teoria, finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de proposições pertencentes à linguagem da LP, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ . Este conjunto representa proposições que devem ser assumidas como verdade no momento do raciocínio.
- Um conjunto finito de hipóteses  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP, denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto representa hipóteses que junto com o conjunto T explicam os fatos representados pelo conjunto F.
- Um conjunto com um único fato  $F = \{ f_i \}$ , um átomo pertencente à linguagem da LP. Este conjunto representa a ocorrência de uma evidência, uma manifestação, um sintoma, uma observação, uma marca, um sinal, etc., a ser explicado pelo raciocínio abduutivo.

□

A realização de um raciocínio abduutivo nesta estrutura inclui: (1) a formulação de um conjunto H de hipóteses que juntamente com um conjunto T (teoria) expliquem o fato do conjunto F e; (2) seguintes condições sejam satisfeitas:

$$T \not\models F, \quad (1.1)$$

$$T \cup \{ h \} \models F, \forall h \in H, \text{ sendo } H \neq \emptyset, \quad (1.2)$$

$$T \cup \{ h \} \not\models \perp, \forall h \in H, \text{ sendo } H \neq \emptyset, \quad (1.3)$$

$$\{ h \} \not\models F, \forall h \in H, \text{ sendo } H \neq \emptyset. \quad (1.4)$$

A condição da expressão (1.1) previne que o conjunto T sozinho tenha como consequência lógica o conjunto de fatos, i.e., que o conjunto T explique por si só o fato observado, sem necessidade de hipóteses.

Hipóteses satisfazendo a condição (1.2) são classificadas de *hipóteses candidatas* e explicam o fato denotado por F. Hipóteses candidatas satisfazendo a condição (1.3) ao unirem-se à teoria não geram uma contradição e são classificadas *hipóteses consistentes*. Hipóteses consistentes atendendo a condição (1.4) são classificadas de *hipóteses explanatórias*, hipóteses que não explicam o fato por si só, necessitando de uma teoria para explicar os fatos. Hipóteses inconsistentes e não explanatórias, i.e., que não atendem as condições (1.3) e (1.4), devem ser descartadas.

**Exemplo 2.12.** Utilizando o Cenário 1.1, constitui-se os conjuntos T, H e F da seguinte maneira:

**Teoria:**

- $T = \{ \textit{choveu} \rightarrow \textit{gramadoMolhado}, \textit{irrigadoresLigados} \rightarrow \textit{gramadoMolhado} \}$ .

**Fatos:**

- $F = \{ \textit{gramadoMolhado} \}$ .

O conjunto Teoria T e o conjunto de fatos F satisfazem a condição (1.1). Em contrapartida, se a teoria fosse  $T_1 = \{ \textit{choveu} \rightarrow \textit{gramadoMolhado}, \textit{irrigadoresLigados} \rightarrow \textit{gramadoMolhado}, \textit{choveu} \}$  a condição (1.1) não seria satisfeita pois  $T_1$  possui  $\perp$  como consequência lógica.

Seja  $H = \{ \textit{choveu}, \textit{irrigadoresLigados}, \textit{choveu} \wedge \textit{irrigadoresLigados}, \textit{choveu} \wedge \neg \textit{gramadoMolhado}, \textit{gramadoMolhado} \}$  um conjunto de hipóteses candidatas. Cada hipótese  $h \in H$  satisfaz a condição (1.2), e cada hipótese  $h \in \{ \textit{choveu}, \textit{irrigadoresLigados}, \textit{choveu} \wedge \textit{irrigadoresLigados}, \textit{gramadoMolhado} \}$  é consistente, satisfazendo a condição (1.3). A hipótese  $\{ \textit{choveu} \wedge \neg \textit{gramadoMolhado} \}$  é inconsistente, pois  $T \wedge \{ \textit{choveu} \wedge \neg \textit{gramadoMolhado} \} \models \perp$ , devendo ser descartada. As hipóteses  $h \in \{ \textit{choveu}, \textit{irrigadoresLigados}, \textit{choveu} \wedge \textit{irrigadoresLigados} \}$  são explanatórias, pois satisfazem a condição (1.4), no entanto a hipótese  $\{ \textit{gramadoMolhado} \}$  é não explanatória, pois  $\{ \textit{gramadoMolhado} \} \models \perp$  devendo ser descartada. Assim, removendo as hipóteses inconsistentes e não explanatórias temos um conjunto de hipóteses  $H = \{ \textit{choveu}, \textit{irrigadoresLigados}, \textit{choveu} \wedge \textit{irrigadoresLigados} \}$ .

□

Além da utilização da estrutura THF e das condições (1.1), (1.2), (1.3) e (1.4), alguns algoritmos aplicam algum critério extra lógico para a seleção de boas hipóteses, como discutido na Seção 1.4.

### 2.3 – Abordagens em Lógica para realização do raciocínio abduutivo

Dillig & Dillig (2013) apresentam uma ferramenta baseada na estrutura THF, denominada EXPLAIN, para realizar raciocínios abdutivos. Nesta abordagem também são aplicados critérios extra lógicos referentes à *simplicidade* e *generalidade* de hipóteses. De forma geral EXPLAIN computa as possíveis hipóteses que explicam a maior quantidade de fatos, seleciona dentre todas as mais simples, i.e., as hipóteses que explicam os fatos com menor quantidade de proposições, e das hipóteses mais simples, as com maior generalidade.

Aliseda (2006) apresenta uma abordagem lógica para abdução. Usando Lógica Proposicional e também Lógica de Predicados de Primeira Ordem, Aliseda estabelece um sistema de raciocínio abduutivo para formular as melhores hipóteses H para explicar um fato atômico do conjunto F considerando um conjunto T. As hipóteses abdutivas são obtidas através de uma adaptação do Tableaux Semântico (um método utilizado para a prova de teoremas). Aliseda em seu estudo aplica o conceito de *simplicidade* para a seleção das melhores hipóteses abdutiva.

Outra abordagem abdutiva em lógica é a Programação Lógica Abdutiva – *Abductive Logic Programming* – ALP (Kakas, Kowalski & Toni, 1995). ALP estende a Programação Lógica (Nilsson & Maluszynski, 1995) tradicional por permitir a declaração de predicados especiais, chamados predicados abduíveis. Estes predicados são utilizados na formulação de hipóteses para a realização de raciocínios abdutivos.

O sistema de raciocínio abduutivo é descrito por um programa que descreve uma teoria T, tal como um programa em Prolog, um conjunto A de predicados abduíveis sobre os quais as hipóteses abdutivas H são formuladas, um conjunto de regras de restrição de integridade IC, usadas como filtros na formulação de hipóteses, e os fatos F, como consultas da Programação Lógica.

O Apêndice II descreve, com algum nível de detalhe, algoritmos com base em Lógica para a realização de raciocínios abdutivos.

## **2.4 – Desvantagens das abordagens existentes**

Uma das desvantagens das abordagens de Aliseda (2006) e Dillig & Dillig (2013), está na restritividade da estrutura de raciocínio proposta (THF). Na abordagem de Aliseda o raciocínio abduativo se dá por apenas um único fato, o que restringe excessivamente a realização do raciocínio. Em situações do cotidiano observa-se em geral não apenas um, mas vários fatos, sendo assim uma definição mais abrangente seria a que contemplasse um conjunto de fatos observados. Em ambas, outra desvantagem é a falta de uma maneira de explicitar condições sem a “contaminação” e descaracterização dos conjuntos T e F; nosso entendimento sobre este assunto é abordado no Capítulo 3.

Com relação a este último aspecto, Thagard (1978), em um ensaio filosófico sobre critérios de seleção da melhor explicação, no qual investiga o fenômeno da abdução em descobertas científicas, argumenta que uma estrutura para raciocínio abduativo deve conter um conjunto de condições aceitas que opere junto com um conjunto de teorias para possibilitar a formulação de hipóteses a partir de fatos observados.

Na abordagem de Aliseda é utilizado o Tableaux Semântico ao invés do mecanismo de Resolução que, segundo Folkler (2002), possui uma diferença positiva de performance em relação ao Tableaux Semântico.

A abordagem de Programação Lógica Abduativa (ALP), a de Dillig & Dillig, se distingue da abordagem de Aliseda principalmente por utilizar um conjunto de fatos em vez de um único fato. Na abordagem da ALP é permitido estabelecer restrições de integridade à realização do raciocínio. Provida pela introdução das restrições de integridade na estrutura de raciocínio, a ALP permite delimitação da formulação de hipóteses abduativas, o que contribui para a diminuição do tempo de execução subjacente ao processo. Tendo objetivo delimitativo, as restrições de integridade da ALP têm também a desvantagem de não permitir, de forma ampla, a modelagem na estrutura de raciocínio de contextos, circunstâncias, intenções etc. (condições aceitas, que serão discutidas no Capítulo 3). A necessidade de predefinir possíveis hipóteses abduativas parece também ter sido idealizada na ALP para administrar o tempo de execução, entretanto, produz como efeito outra desvantagem, a necessidade de previamente descrever possíveis hipóteses.

Uma desvantagem comum às abordagens estudadas é a pequena atenção dada à seleção de boas hipóteses, entre várias que podem ser formuladas. A abordagem de Aliseda restringe-se ao critério de simplicidade sintática de hipóteses, a de Dillig & Dillig ao critério de generalidade e também o de simplicidade sintática e a abordagem da Programação Lógica Abdutiva é omissa quanto a este assunto.

Contrasta com isto a convicção deste trabalho de que uma abordagem para a realização de raciocínios abdutivos deve levar em conta critérios de seleção de boas hipóteses e a corroboração da investigação deste elemento do processo abduutivo por diversos pesquisadores em campos como a Filosofia (Harman, 1965; Thagard, 1978, 1989), Psicologia (Magnani, 2009) e a aplicação às Ciências Médicas (Josephson & Josephson, 1994).

## **2.5 – Objetivo**

Considerando as desvantagens identificadas nas abordagens estudadas, o objetivo deste trabalho é a proposta de uma abordagem para realização de raciocínios abdutivos que:

- Utilize uma estrutura de raciocínio que:
  - Considere um ou mais fatos observados;
  - Explicitar contextos, circunstâncias e intenções, i.e., condições aceitas que operem junto com um conjunto de teorias;
  - Permita a formulação de hipóteses sem a necessidade de predefinir possíveis candidatas.
- Considere critérios para a seleção de boas hipóteses.

Objetiva-se também o desenvolvimento e implementação de um algoritmo abduutivo, a ser denominado Peirce, que consolidará a proposta e, se possível, a traduza com a utilização de mecanismos eficientes que reduzam o tempo de execução do algoritmo.

## **2.6 – Metodologia**

Para a proposta de uma nova estrutura de raciocínio que atendessem aos objetivos delineados na Seção 2.5, propõe-se a análise de diferentes situações de uso de raciocínio

abduativo. Em primeiro lugar, buscou-se em diferentes mídias cenários que demandavam raciocínio abduativo, entre elas:

- Estórias da Literatura Policial, de Mistério e Suspense;
- Pinturas abstratas e concretas;
- Vídeos em TV, YouTube e DVD;
- Sons capturados em DVD e no YouTube;
- Textos em jornais e revistas;
- Descobertas científicas descritas em artigos;
- Projetos de engenharia descrevendo sistemas;
- Jogos de computador.

Diversos cenários foram selecionados nestas mídias e de observações do cotidiano. Tais cenários serviram para permitir a investigação num rico espaço que incluiu:

- Situações do cotidiano envolvendo diálogos, congestionamento de trânsito e tempo;
- Estórias das *Aventuras de Sherlock Holmes – A liga dos cabeças vermelhas* (Doyle, 2007) e de *O Seminarista* de Fonseca (2009);
- Pinturas concretas e abstratas como *Skull and Pitcher* de Pablo Picasso;
- Vídeos e filmes em TV e no YouTube abordando acusação e defesa em julgamentos e investigação policial, como, por exemplo, o documentário da TV A&E sobre O Caso Richthofen, que inspirou um cenário discutido no Apêndice I.
- Diagnóstico de doenças;
- Interpretação de textos de literários, músicas, piadas, poemas etc.

A Seção 1.3 do Capítulo 1 e o Apêndice I descrevem alguns cenários investigados.

Os cenários foram investigados tendo em vista a adequação de estruturas de raciocínio conhecidas, explicitando suas dificuldades em modelar raciocínios abduativos.

Paralelamente, buscou-se na literatura científica, investigando textos de áreas como a Filosofia, a Psicologia e Inteligência Artificial, sustentação para a estrutura de raciocínio proposta, como, por exemplo, em (Thagard, 1978) encontra-se sustentação para a

proposta de uma estrutura diferente da estrutura THF, uma estrutura que inclua um conjunto a mais representando condições.

Alguns autores também parecem reforçar a ideia de que a realização de raciocínios abdutivos é dependente de algum tipo de condição, por exemplo, em uma tradução, Doyle (2014) argumenta: “Um dos clássicos exemplos é o de explicar a grama molhada. Se a grama está molhada, provavelmente choveu. Chuva é a melhor explicação para grama molhada, especialmente em New England, onde viveu Peirce, mas não necessariamente a melhor explicação no Arizona na alta temporada da estação de secas, onde sistemas automatizados de irrigadores podem ser a melhor explicação para a grama molhada – especialmente se a grama está molhada, mas a rua está seca.”.

A estrutura de raciocínio da proposta pode ser instanciada, entre outras possibilidades, por meio de alguma Lógica, como um problema de busca, raciocínio probabilístico sobre Redes Bayesianas. Optou-se pela Lógica e, em particular, pela Lógica Proposicional, pela sua simplicidade, não necessidade de mensuração de parâmetros probabilísticos, muitas vezes difíceis de serem obtidos e definidos, e proximidade semântica com os conceitos de raciocínio e inferência lógica são classicamente tratados.

O objetivo de definir critérios para a seleção de boas hipóteses abdutivas tem sido tratado no Campo da Filosofia (Thagard, 1978, 1989) e Teorias Clássicas da Psicologia Cognitiva (Magnani, 2009). Este trabalho fez uma diferente opção metodológica. Quis-se entender que os critérios para seleção de boas hipóteses são um fator humano e, como tal, devem ser pesquisado por meio de métodos experimentais. Para tanto, propõe-se uma Pesquisa Experimental envolvendo pessoas a fim de verificar a coincidência da formulação de hipóteses abdutivas feitas pelo algoritmo Peirce.

O algoritmo Peirce fará uso do mecanismo de resolução o qual, segundo Folkler (2002), apresenta uma diferença positiva no tempo de execução computacional em relação ao Tableaux Semântico que foi utilizado na abordagem lógica de Aliseda (2006).

Por fim, há a opção metodológica de implementar o algoritmo Peirce em um ambiente Web, para facilitar a sua distribuição.

# Capítulo 3

## Estrutura Quaternária TCHF e um Critério para Seleção de Boas Hipóteses

*Este capítulo apresenta o conceito de Condições Aceitas, a estrutura quaternária TCHF para a formalização de raciocínios abduativos e um critério para a seleção de boas hipóteses. A Seção 3.1 apresenta o entendimento deste trabalho sobre uma característica dos raciocínios abduativos, aqui denominada, Condições Aceitas. A Seção 3.2 apresenta a estrutura TCHF, uma extensão da estrutura para formalização de raciocínio THF, a qual inclui o conjunto  $C$  representando Condições Aceitas. A Seção 3.3 apresenta um critério para a seleção para boas hipóteses composto por força explanatória e complexidade de hipóteses.*

### 3.1 – Condições Aceitas

Raciocinar condicionalmente é uma atividade comum; estar sujeito a condições que devem ser consideradas no momento da realização do raciocínio faz parte do cotidiano das pessoas.

Raciocínios abduativos podem gerar diferentes hipóteses dependendo das condições em que o realizador do raciocínio se encontra. Estas condições podem incluir contextos (referente a espaço), circunstâncias (referente a tempo), intenções, fé, crenças, etc.

Exemplos de condições específicas usadas em raciocínios abduativos são: (1) No raciocínio abduativo usado para diagnósticos médicos, contextos regionais podem ser utilizados para especificar um conjunto de doenças que são mais comuns ou menos comuns de acordo com uma região específica; (2) No raciocínio abduativo usado para estudo antropológicos, especificar possíveis agentes causadores da morte de um homínido baseado no conhecimento de que o homínido viveu há cerca de 4 milhões de anos (circunstância); (3) No raciocínio abduativo usado para interpretações jurídicas, possíveis condições devem ser especificadas com a intenção de absolver (ou condenar) um réu; e (4) No raciocínio abduativo usado para crenças de caráter metafísico e religioso, a crença de que existe vida após a morte pode ser declarada como uma condição na qual o raciocínio é feito.

A ideia de incluir condições ao raciocínio abduativo nos chegou através de duas distintas fontes, uma intuitiva, na medida em que as primeiras investigações foram feitas, e a segunda por encontrar, em Thagard (1978, 1989) e Peirce (1934), suporte filosófico para isto. Thagard (1978) argumenta a necessidade de um conjunto de condições que operem junto aos outros conjuntos para a realização do raciocínio abduativo. Neste sentido, este trabalho entende o raciocínio abduativo não somente como um raciocínio geral de fatos para hipóteses, mas também como um tipo de raciocínio adaptável à situação que está sendo imposta a ele. A seguir apresentamos um exemplo que propõe realizar um raciocínio abduativo remetendo-se a diferentes condições.

**Exemplo 3.1.** Tendo por teoria que a travessia entre as cidades A e B pode ser feita de diversas maneiras:

- *A pé.*

- *A cavalo.*
- *Por automóvel.*
- *Por trem.*
- *Por avião.*

Se tem-se como fato observado a *travessia da Cidade A para a Cidade B*, pode-se abduzir que as possíveis maneiras de se fazer essa travessia sejam as citadas acima.

A partir da situação exemplo a Tabela 3.1 exemplifica por meio de condições contextuais, circunstanciais e intencionais, como a formulação de hipóteses é afetada pelas condições na realização do raciocínio abduzitivo.

**Tabela 3.1: Exemplo de condições contextual, circunstancial e intencional aplicadas ao raciocínio abduzitivo.**

<i>Contextual</i>	<i>Circunstancial</i>	<i>Intencional</i>
<p>Aceitando-se a condição contextual de que a distância entre a Cidade A e a Cidade B é muito grande, as possíveis hipóteses passam a ser:</p> <ul style="list-style-type: none"> <li>• <i>Por Avião.</i></li> </ul> <p>O motivo pelo qual a única hipótese ser <i>Por Avião</i> se dá pelo motivo que, comumente, para percorrer grandes distâncias o comum é utilizar este meio de transporte.</p>	<p>Aceitando-se a condição circunstancial de que a travessia está sendo realizada no século XVI, as possíveis hipóteses a serem formuladas passam a ser:</p> <ul style="list-style-type: none"> <li>• <i>A pé.</i></li> <li>• <i>A cavalo.</i></li> </ul> <p>O motivo pelo qual as hipóteses são reduzidas a essas duas é devido ao fato de que no século XVI não haviam automóveis, trens e aviões.</p>	<p>Aceitando-se a condição intencional de que a travessia deve ser feita do modo mais barato possível, as possíveis hipóteses a serem formuladas passam a ser:</p> <ul style="list-style-type: none"> <li>• <i>A pé.</i></li> </ul> <p>Dentre todas, essa é a hipótese à qual não há custo algum com transporte.</p>

□

### 3.2 – Estrutura Quaternária TCHF

Este trabalho propõe uma estrutura quaternária para formalização do raciocínio abduativo, denominada TCHF. A estrutura de raciocínio TCHF se diferencia da estrutura THF por permitir uma quantidade finita de fatos observados e não somente um e, também, por incluir o conjunto C de Condições Aceitas. O conjunto C tem por função explicitar condições aceitas que permeiam o raciocínio abduativo no momento de sua realização

Considere o Exemplo 3.2 de um raciocínio abduativo na estrutura THF.

**Exemplo 3.2.** José possui em frente de sua casa um grande gramado. Um dia José chega em casa e observa que seu gramado está molhado. Considerando que (1) a *chuva molha o gramado* e que (2) *irrigadores instalados através do gramado molham o gramado*, qual hipótese pode explicar o fato do *gramado estar molhado*?

Formalizando a situação através da estrutura THF temos:

**O conjunto Teoria:**

- $T = \{ \textit{choveu} \rightarrow \textit{gramadoMolhado}, \textit{irrigadoresLigados} \rightarrow \textit{gramadoMolhado} \}$ .

**O conjunto Fatos:**

- $F = \{ \textit{gramadoMolhado} \}$ .

□

Se a este exemplo for introduzida uma condição contextual de que, por exemplo, *o tanque de abastecimento dos irrigadores está sem água*, e que *se o tanque de abastecimento dos irrigadores está sem água então os irrigadores não estão ligados*, esta condição deve ser explicitada nos conjuntos T ou F.

Se esta condição for explicitada no conjunto T, “contamina-se” o conjunto com informações que não deveriam fazer parte do conjunto teoria. O conjunto T deve ser utilizado para encontrar hipóteses que expliquem os fatos em qualquer situação imposta. Incluir este tipo de informação, que representa condições específicas de uma situação, faz com que o conjunto T fique enviesado a explicar somente determinadas situações, i.e., fazendo com que a teoria perca sua generalidade, assim como apresenta o Exemplo 3.3.

**Exemplo 3.3.** Considerando o Exemplo 3.2 e incluindo a seguinte condição à estrutura THF, *o tanque de abastecimento dos irrigadores está sem água; e que se o tanque de abastecimento dos irrigadores está sem água então os irrigadores não estão ligados*, temos:

**Novo conjunto Teoria:**

- $T = \{ \text{choveu} \rightarrow \text{gramadoMolhado}, \text{irrigadoresLigados} \rightarrow \text{gramadoMolhado}, \text{tanqueSemAgua}, \text{tanqueSemAgua} \rightarrow \neg \text{irrigadoresLigados} \}$ .

□

Podemos notar que o conjunto T do Exemplo 3.3 deixa de ser um conjunto geral que serviriam para explicar o fato do *gramado estar molhado*, pois a partir dele, em qualquer situação em que apareça o fato *gramadoMolhado*, pode-se concluir apenas a hipótese *choveu*.

Ao conjunto F que representa somente os fatos a serem explicados, não seria correto a inclusão condições, pois condições não são fatos que se queira explicar. Vale ressaltar que algumas abordagens que utilizam a estrutura THF utilizam o conjunto F contendo somente um fato a ser explicado, impossibilitando ainda mais o uso das condições juntamente ao conjunto F.

Este trabalho propõe uma abordagem para explicitar condições aceitas ao raciocínio abduativo, sem que haja a descaracterização dos conjuntos T e F. A maneira com a qual este trabalho propõe explicitar estas condições é a partir de um conjunto de Condições Aceitas (C). Os elementos deste conjunto são utilizados na realização do raciocínio para garantir que as hipóteses formuladas sejam consistentes, não só à teoria, mas também às condições do conjunto de Condições Aceitas.

**Exemplo 3.4.** Considerando novamente o cenário do Exemplo 3.3, desta vez utilizando a estrutura TCHF.

**Teoria:**

- $T = \{ \text{choveu} \rightarrow \text{gramadoMolhado}, \text{irrigadoresLigados} \rightarrow \text{gramadoMolhado} \}$ .

**Condições Aceitas:**

- $C = \{ \text{tanqueSemAgua}, \text{tanqueSemAgua} \rightarrow \neg \text{irrigadoresLigados} \}$ .

**Fatos:**

- $F = \{ \textit{gramadoMolhado} \}$ .

□

Assim tem-se a estrutura TCHF (Definição 3.1) para formalização de raciocínios abduativos que adicionalmente utiliza a Forma HF especificada na Definição 2.16.

**Definição 3.1 (Estrutura TCHF).** A estrutura TCHF para raciocínio abduativo é um sistema  $\langle T, C, H, F \rangle$  que consiste em:

- Um conjunto finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ . Este conjunto representa as proposições que devem ser assumidas como verdadeiras no momento do raciocínio abduativo.
- Um conjunto finito,  $C = \{ c_1, c_2, c_3, \dots, c_p \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_p$ . Este conjunto representa condições que devem ser assumidas como verdadeiras no momento do raciocínio abduativo.
- Um conjunto finito,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto representa as hipóteses que juntamente à teoria (T) e às condições aceitas (C), explicam os fatos observados (F).
- Um conjunto finito e não vazio,  $F = \{ f_1, f_2, f_3, \dots, f_q \}$ , de átomos pertencentes à linguagem LP, denotando  $f_1 \wedge f_2 \wedge f_3 \wedge \dots \wedge f_q$ . Este conjunto representa os fatos que devem ser explicados através do raciocínio abduativo; estes fatos incluem evidências, sintomas, observações, marcas, sinais, etc.

□

As restrições impostas pela Forma HF foram feitas com o intuito de diminuir o tempo de execução do problema de realizar um raciocínio abduativo em Lógica. O problema de realizar um raciocínio abduativo em Lógica foi provado pertencer à classe NP-Completo (Eiter & Gottlob, 1995), por meio da restrição das proposições à Forma HF, as quais são facilmente convertidas em Cláusulas Horn (Howard, 1999), é possível desenvolver um algoritmo polinomial para resolver o problema de abdução em Lógica.

A realização de um raciocínio abduutivo, na estrutura TCHF, inclui: (1) a formulação de um conjunto de hipóteses  $H$  que juntamente com uma teoria  $T$  e com condições aceitas  $C$  explique, totalmente ou parcialmente, os fatos do conjunto  $F$  e; (2) as seguintes condições sejam satisfeitas:

$$T \cup C \not\models F, \quad (3.1)$$

$$T \cup C \cup \{ h \} \models_p F, \forall h \in H, \text{ sendo } H \neq \emptyset, \quad (3.2)$$

$$T \cup C \cup \{ h \} \not\models \perp, \forall h \in H, \text{ sendo } H \neq \emptyset. \quad (3.3)$$

A condição (3.1) garante que a união  $T \cup C$  seja geral, i.e., não explica os fatos do conjunto  $F$ , e que o conjunto  $C$  não possui condições que, em conjunção com  $T$ , impossibilitem a formulação de hipóteses que expliquem  $F$ .

Hipóteses satisfazendo a condição (3.2) são chamadas *hipóteses candidatas*, e explicam um ou mais fatos presentes no conjunto  $F$ . Hipóteses candidatas que satisfazem a condição (3.3) são *hipóteses consistentes*, i.e., hipóteses que, quando unidas aos conjuntos  $T$  e  $C$ , não resulte em uma contradição.

A condição (3.2) utiliza um conceito de consequência lógica parcial especificado na Definição 3.2.

**Definição 3.2** ( $A \models_p B$ , i.e.,  $B$  é uma consequência lógica parcial de  $A$ ). Sejam  $A = \{ a_1, a_2, a_3, \dots, a_n \}$ ,  $B = \{ b_1, b_2, b_3, \dots, b_m \}$  e  $C = \{ c_1, c_2, c_3, \dots, c_q \}$ , três conjuntos finitos e não vazios de proposições pertencentes à linguagem da LP, em que  $C \subseteq B$ ,  $A \models_p B$ , se e somente se,  $A \models C$ .

□

### 3.3 – Critério para a Seleção de Boas Hipóteses.

Diversas hipóteses podem ser formuladas durante a realização de raciocínios abdutivos. Assim, no raciocínio abduutivo existe um processo de formular  $m \geq 1$  hipóteses gerais, seguido da seleção de  $n \leq m$  boas hipóteses.

Hipóteses exibem propriedades tais como força explanatória e complexidade, conforme definidos a seguir.

**Definição 3.3 (Força Explanatória):** A força explanatória de uma hipótese é a relação entre o número de fatos que ela explica e a quantidade de fatos presente no conjunto de fatos. Por exemplo, se uma hipótese explica 4 fatos dos 5 encontrados no conjunto de fatos, a força explanatória desta hipótese é de 4/5; hipóteses que explicam todos os fatos do conjunto de fatos observados têm sua força explanatória igual a 1.

□

A Figura 3.1 apresenta algumas situações de hipóteses e suas forças explanatórias.

Situação 1		Fatos					
Hipóteses		f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	
h <sub>1</sub>		√	√	√	√		
h <sub>2</sub>			√				
h <sub>3</sub>		√				√	

(a)

Situação 2		Fatos					
Hipóteses		f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	
h <sub>1</sub>		√					
h <sub>2</sub>			√				
h <sub>3</sub>		√					

(b)

Situação 3		Fatos					
Hipóteses		f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	
h <sub>1</sub>		√	√	√	√		
h <sub>2</sub>			√	√	√	√	
h <sub>3</sub>				√			

(c)

Situação 4		Fatos				
Hipóteses		f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	
h <sub>1</sub>		√	√			
h <sub>2</sub>				√	√	
h <sub>3</sub>			√			

(d)

**Figura 3.1:** Situações exemplo de força explanatória de hipóteses (O “√” representa que a hipótese explica o fato observado).

Na situação 1 observa-se que a hipótese  $h_1$  tem maior força explanatória do que as hipóteses  $h_2$  e  $h_3$ . Na situação 2 as três hipóteses tem a mesma força explanatória. A situação 3 também apresenta as hipóteses  $h_1$  e  $h_2$  com mesma força explanatória porém explicando conjuntos diferentes de fatos. Na situação 4 as hipóteses  $h_1$  e  $h_2$  tem a mesma força explanatória, porém explicam diferentes conjuntos disjuntos fatos.

**Definição 3.4 (Complexidade):** A complexidade de uma hipótese é definida como o número de literais que a descrevem, independente dos conectivos lógicos contidos na

proposição. Por exemplo, uma hipótese  $p$  tem a complexidade igual a 1 e uma hipóteses  $p \wedge q \wedge r$  tem complexidade igual a 3.

□

A Figura 3.2 apresenta um conjunto  $H$  contendo quatro hipóteses juntamente com suas complexidades.

Conjunto de hipóteses $H = \{ h_1, h_2, h_3, h_4 \}$ : Complexidade:	
$h_1 = a.$	1
$h_2 = a \wedge b.$	2
$h_3 = a \wedge b \wedge c.$	3
$h_4 = a \wedge b \wedge c \wedge d.$	4

**Figura 3.2:** Um conjunto  $H$  de quatro hipóteses e a complexidade das hipóteses.

Este trabalho utiliza os conceitos de força explanatória e complexidade para estabelecer um critério para a seleção de boas hipóteses na Definição 3.5.

**Definição 3.5 (Critério para a Seleção de Boas Hipóteses).** Dado um conjunto  $H$  de hipóteses candidatas para explicar um conjunto  $F$  de fatos,  $h \in H$  é considerada uma boa hipótese se satisfaz às seguintes condições:

- A força explanatória de  $h$  é maior ou igual a uma constante  $\lambda_1$ .
- A complexidade de  $h$  é menor ou igual a uma constante  $\lambda_2$ .
- A hipótese  $h$  possui complexidade mínima entre todas as hipóteses que tem a força explanatória máxima em  $H$ .

□

Nos experimentos realizados nesta dissertação utilizamos os valores  $\lambda_1 = 0.5$  e  $\lambda_2 = 5$ . Os valores de  $\lambda_1$  e  $\lambda_2$ , utilizados no experimento, foram escolhidos para coincidir com fatores humanos. Considerando o experimento de Miller (Mackenzie, 2013), a memória humana e a capacidade de processamento humano estão limitadas a  $7 \pm 2$  elementos

simultâneos, assim  $\lambda_2 = 5$ . Boas hipóteses explicam pelo menos 50% dos fatos observados, assim  $\lambda_1 = 0.5$ .

O Exemplo 3.5 apresenta uma utilização do Critério para a Seleção de Boas Hipóteses,

**Exemplo 3.5.** Considerando:

**Os conjuntos T, C e F:**

- $T = \{ pneumonia \rightarrow febre \wedge dor\_de\_cabeca \wedge coriza, gripe \rightarrow febre \wedge coriza, rinite \rightarrow dor\_de\_cabeca \wedge coriza \}$ .
- $C = \{ \}$ .
- $F = \{ febre, dor\_de\_cabeca, coriza \}$ .

**Hipóteses consistentes:**

- $H = \{ pneumonia, gripe, rinite, pneumonia \wedge gripe, pneumonia \wedge rinite, gripe \wedge rinite, pneumonia \wedge gripe \wedge rinite \}$ .

A Tabela 3.2 sumariza as forças explanatórias e complexidades das hipóteses consistentes.

**Tabela 3.2: Força explanatória e complexidade das hipóteses do Exemplo 3.5.**

<i>Hipóteses</i>	<i>Fatos</i>			<i>Força Explanatória</i>	<i>Complexidade</i>
	<i>febre</i>	<i>dor_de_cabeca</i>	<i>coriza</i>		
<i>pneumonia</i>	√	√	√	1	1
<i>gripe</i>	√		√	0,66	1
<i>rinite</i>		√	√	0,66	1
<i>pneumonia</i> $\wedge$ <i>gripe</i>	√	√	√	1	2
<i>pneumonia</i> $\wedge$ <i>rinite</i>	√	√	√	1	2
<i>gripe</i> $\wedge$ <i>rinite</i>	√	√	√	1	2
<i>pneumonia</i> $\wedge$ <i>gripe</i> $\wedge$ <i>rinite</i>	√	√	√	1	3

Todas as hipóteses possuem força explanatória maior que  $\lambda_1 = 0,5$  e complexidade menor que  $\lambda_2 = 5$ . As hipóteses *pneumonia*, *pneumonia*  $\wedge$  *gripe*, *pneumonia*  $\wedge$  *rinite*,

*gripe*  $\wedge$  *rinite*, *pneumonia*  $\wedge$  *gripe*  $\wedge$  *rinite* possuem força explanatória igual a 1, o máximo dentre todas as hipóteses, no entanto *pneumonia* possui a menor complexidade entre elas, fazendo com que *pneumonia* seja selecionada como boa hipótese.

□

# Capítulo 4

## Algoritmo Peirce em Lógica Proposicional

*Este capítulo propõe e discute o algoritmo para a realização de raciocínios abduativos em Lógica Proposicional denominado Peirce, em homenagem ao filósofo americano Charles Sanders Peirce, que estabeleceu o conceito de raciocínio abduativo. A Seção 4.1 detalha como o algoritmo faz uso da estrutura TCHF e o Critério para a Seleção de Boas Hipóteses, definidos ambos no Capítulo 3, e apresenta o seu pseudocódigo. A Seção 4.2 detalha o algoritmo Peirce. A Seção 4.3 trata do tempo de execução do algoritmo Peirce. Por fim a Seção 4.4 trata da Pesquisa Experimental, que tem como objetivo verificar a tendência das soluções computadas pelo algoritmo Peirce de coincidirem com o raciocínio abduativo realizado por pessoas.*

## 4.1 – Algoritmo Peirce em Lógica Proposicional

O algoritmo  $\text{Peirce}(T, C, H)$  realiza um raciocínio abdutivo. A Figura 4.1 apresenta o pseudocódigo do algoritmo Peirce.

```
Algoritmo Peirce(T, C, F):  
Entrada:  
• Um conjunto finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ . Este conjunto representa as proposições que devem ser assumidas como verdadeiras no momento do raciocínio abdutivo.  
• Um conjunto finito,  $C = \{ c_1, c_2, c_3, \dots, c_p \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_p$ . Este conjunto representa condições que devem ser assumidas como verdadeiras no momento do raciocínio abdutivo.  
• Um conjunto finito e não vazio,  $F = \{ f_1, f_2, f_3, \dots, f_q \}$ , de átomos pertencentes à linguagem LP, denotando  $f_1 \wedge f_2 \wedge f_3 \wedge \dots \wedge f_q$ . Este conjunto representa os fatos que devem ser explicados através do raciocínio abdutivo; estes fatos incluem evidências, sintomas, observações, marcas, sinais, etc.  
Saída:  
• Um conjunto finito,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto representa as hipóteses que juntamente à teoria (T) e às condições aceitas (C), explicam os fatos observados (F).  
1 {  
2   se Consistentes(T, C) então  
3   {  
4     R := Forma_Normal_Conjuntiva_Clausulas_Horn(T, C,  
5       ¬F);  
6     R := Resolucao(R);  
7     se existe alguma cláusula vazia em R então  
8       escreva ("Não há hipóteses a serem formuladas,  
9         pois  $T \cup C \models F$ ");  
10    senão  
11    {  
12      H := Formular_Hipoteses_Candidatas(R);  
13      H := Remover_Hipoteses_Inconsistentes(T, C, H);  
14      H := Selecionar_Boas_Hipoteses(T, C, H, F);  
15      retornar H.  
16    }  
17  }  
18  senão  
19    escreva ("Impossível formular hipóteses. A Teoria e  
20      as Condições Aceitas formam um sistema  
21      inconsistente.");  
22 }
```

Figura 4.1: Algoritmo Peirce para realização de raciocínios abduativos.

O algoritmo utiliza a estrutura TCHF (Definição 3.1) e o Critério de Seleção de Boas Hipóteses (Definição 3.5) para formular boas hipóteses que expliquem, totalmente ou parcialmente, um conjunto de fatos (F) de acordo com uma teoria (T) e condições aceitas (C) tidos como entrada do algoritmo. As hipóteses formuladas pelo algoritmo Peirce satisfazem as condições das expressões (3.1), (3.2), (3.3).

Resumidamente o algoritmo  $\text{Peirce}(T, C, F)$  formula hipóteses candidatas e as guarda no conjunto H (linha 10) para em seguida, remover as hipóteses inconsistentes de H (linha 11) e, então, selecionar e deixar somente boas hipóteses em H (linha 12) de acordo com o Critério para a Seleção de Boas Hipóteses.

Primeiramente é testada a consistência de  $T \cup C$  (linha 2). Entende-se por inconsistência de  $T \cup C$  se  $T \cup C \models \perp$ . Em uma eventual inconsistência de  $T \cup C$  não há como formular hipóteses através destes conjuntos, uma vez que  $\perp \models V$  e também  $\perp \models F$ , ou seja, através de uma inconsistência são derivadas hipóteses inconsistentes que têm como consequência lógica qualquer proposição.

Para computar hipóteses candidatas, i.e., hipóteses que atendem a condição (3.2), o algoritmo traduz o conjunto de proposições  $T \cup C \cup \neg F$  para a FNC com cláusulas Horn<sup>2</sup> (linha 4) e em seguida aplica o algoritmo  $\text{Resolucao}(R)$  (linha 5). O resultado é armazenado em uma estrutura de dados R (conjunto de cláusulas). Se R não contém cláusulas vazias, então  $T \cup C \not\models F$ , a condição (3.1) é satisfeita e hipóteses candidatas podem ser formuladas (linhas 6, 7 e 8).

Na (linha 10) cada cláusula em R representa uma possível hipótese. Estando R na FNC, negar cada cláusula resulta em uma hipótese candidata, i.e., que atendem a condição (3.2). O algoritmo  $\text{Formular\_Hipoteses\_Candidatas}(R)$  (linha 10) opera da seguinte maneira: recebe um conjunto de proposições resolvidas através do algoritmo  $\text{Resolucao}(R)$  e opera da seguinte maneira: (1) nega cada cláusula em R, obtendo assim as  $m$  primeiras hipóteses candidatas  $h_1, h_2, h_3, \dots, h_m$ , (2) combina estas  $m$  primeiras hipóteses candidatas duas a duas, três a três, etc. até  $\lambda_2$  a  $\lambda_2$ , i.e., gera  $h_i \wedge h_j$  ( $i \neq j$ ) na primeira iteração,  $h_i \wedge h_j \wedge h_k$  ( $i \neq j \neq k$ ) na segunda iteração e  $h_i \wedge h_j \wedge \dots \wedge h_q$  ( $i \neq j \neq \dots \neq$

---

<sup>2</sup> A estrutura TCHF restringe a escrita de proposições à Forma HF (Definição X.X). Isto promove uma certa facilidade de conversão de proposições em cláusulas Horn que permite a diminuição do tempo de execução do problema de realizar um raciocínio abduutivo em Lógica.

$q$ ) na última iteração onde  $q$  é o mínimo entre  $m$  e  $\lambda_2$  ( $\lambda_2$  refere-se à máxima complexidade admitida para as hipóteses, nesta dissertação  $\lambda_2 = 5$ ).

Das hipóteses candidatas são retiradas as hipóteses inconsistentes, i.e., que não atendem a condição (3.3) através do algoritmo `Remover_Hipotese_Inconsistentes(T, C, H)` (linha 11) operando da seguinte maneira, para cada  $h \in H$ : (1) o algoritmo traduz as proposições compostas por  $T \cup C \cup \{ h \}$  para a FNC com cláusulas Horn, (2) aplica a resolução ao sistema resultante de proposições e (3) se uma cláusula vazia estiver presente no resultado da resolução a hipótese  $h$  é retirada de  $H$ , pois a hipótese é inconsistente com  $T$  e  $C$ .

Após a remoção das hipóteses inconsistentes seleciona-se hipóteses consideradas boas de acordo com o Critério para a Seleção de Boas Hipóteses por meio do algoritmo `Selecionar_Boas_Hipotese(T, C, H, F)` (linha 12) que opera da seguinte maneira: (1) computa a força explanatória e a complexidade de cada hipótese  $h \in H$ , (2) remove as hipóteses  $h$  cuja força explanatória seja menor que a constante  $\lambda_1$ , (3) computa o conjunto  $E$  que contem aquelas hipóteses de  $H$  com maior força explanatória, (4) computa o conjunto  $X$  contendo as hipóteses de  $E$  com a menor complexidade e; (5) retorna  $X$  como resposta.

**Exemplo 4.1.** Este exemplo apresenta uma visão geral do funcionamento do algoritmo Peirce, nas seções seguintes o algoritmo é apresentado com maior riqueza de detalhes. Para este exemplo utilizaremos um circuito elétrico simples constituído de uma bateria, um interruptor, uma lâmpada e fios.

O algoritmo Peirce recebe como entrada um conjunto teoria contendo proposições que podem ser utilizadas para explicar o que leva a lâmpada a estar apagada.

**Teoria:**

- $T = \{ \text{interruptor\_desligado} \rightarrow \text{lampada\_apagada}, \text{bateria\_descarregada} \rightarrow \text{lampada\_apagada}, \text{fios\_rompidos} \rightarrow \text{lampada\_apagada} \}$ .

Para o conjunto de condições aceitas vamos assumir, neste exemplo, que os fios utilizados no circuito são novos e que aceitamos por condição que fios novos não estão rompidos.

**Condições Aceitas:**

- $C = \{ fios\_novos, fios\_novos \rightarrow \neg fios\_rompidos \}$ .

E como fato observado temos que a lâmpada está apagada.

**Fatos observados:**

- $F = \{ lampada\_apagada \}$

Sendo  $T \cup C \not\models \perp$ , o algoritmo `Consistentes(T, C)` (linha 2) retorna Verdade, e a estrutura de dados  $R$  é preenchida com  $T \cup C \cup \neg F$  na FNC com cláusulas Horn (linha 4) recebendo o seguinte:

- Na linha 4:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, \neg lampada\_apagada \}$ .
- Na linha 5, após a resolução:  $R = \{ \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos \}$ .

Não havendo cláusulas vazias em  $R$  (testado na linha 6), as hipóteses candidatas são formadas na linha 10 após executar o algoritmo `Formular_Hipotese_Candidatas(R)`:

- Na linha 10:  $H = \{ interruptor\_desligado, bateria\_descarregada, fios\_rompidos, interruptor\_desligado \wedge bateria\_descarregada, interruptor\_desligado \wedge fios\_rompidos, bateria\_descarregada \wedge fios\_rompidos, interruptor\_desligado \wedge bateria\_descarregada \wedge fios\_rompidos \}$ .

Uma vez que  $T \cup C \cup \{ fios\_rompidos \} \models \perp$  então todas as hipóteses que apresentam  $fios\_rompidos$  são inconsistentes sendo assim as hipóteses,  $fios\_rompidos$ ,  $interruptor\_desligado \wedge fios\_rompidos$ ,  $bateria\_descarregada \wedge fios\_rompidos$ ,  $interruptor\_desligado \wedge bateria\_descarregada \wedge fios\_rompidos$ , são removidas de  $H$  pelo algoritmo `Remover_Hipotese_Inconsistentes(T, C, H)` (linha 11), deixando  $H = \{ interruptor\_desligado, bateria\_descarregada, interruptor\_desligado \wedge bateria\_descarregada \}$ . Na linha 12 o algoritmo `Selecionar_Boas_Hipotese(T, C, H, F)` calcula a força explanatória e complexidade de cada hipótese em  $H$ , retornando  $H = \{ interruptor\_desligado, bateria\_descarregada \}$  por serem as hipóteses com maior força explanatória e menor complexidade.

## 4.2 – Detalhamento do algoritmo Peirce

O algoritmo Peirce faz uso de diversos subalgoritmos, esta seção apresenta o pseudocódigo de cada um deles, descreve-os e os exemplifica.

### 4.2.1 – Consistentes(A, B)

O algoritmo `Consistentes(A, B)` (Figura 4.2) recebe dois conjuntos finitos de proposições pertencentes à linguagem da LP na Forma HF e verifica se são consistentes. Ele utiliza os algoritmos `Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C)` (Seção 4.2.2) e `Resolucao(R)` (Seção 4.2.3). Se uma cláusula vazia for derivada pelo algoritmo de resolução significa que a união dos conjuntos é inconsistente.

```

Algoritmo Consistentes(A, B):
Entrada:
  • Conjunto finito  $A = \{ a_1, a_2, a_3, \dots, a_n \}$  de proposições
    pertencentes à linguagem da LP.
  • Conjunto finito  $B = \{ b_1, b_2, b_3, \dots, b_m \}$  de proposições
    pertencentes à linguagem da LP.
Saída:
  • Retorna Verdade caso o conjunto de proposições  $A \cup B$  seja
    consistente. Caso contrário, retorna Falso.
1  {
2    R := Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, {});
3    R := Resolucao(R);
4    se existe alguma cláusula vazia em R então
5      retornar Falso;
6    senão
7      retornar Verdade;
8  }
```

**Figura 4.2: Algoritmo Consistentes (A, B).**

**Exemplo 4.2.** Considerando o cenário do Exemplo 4.1 o algoritmo `Consistentes(A, B)` recebe como entrada os conjuntos  $T = \{ interruptor\_desligado \rightarrow lampada\_apagada, bateria\_descarregada \rightarrow lampada\_apagada, fios\_rompidos \rightarrow lampada\_apagada \}$  e  $C = \{ fios\_novos, fios\_novos \rightarrow \neg fios\_rompidos \}$ , como parâmetros para, respectivamente, A e B, converte  $A \cup B$  para a FNC com cláusulas Horn e aplica o algoritmo de resolução gerando:

- Na linha 2:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \text{fios\_novos}, \neg \text{fios\_novos} \vee \neg \text{fios\_rompidos} \}$ .
- Na linha 3, após resolução:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \}$ .

Não contendo nenhuma cláusula vazia (teste da linha 4) em R, o algoritmo retorna Verdade, indicando que  $A \cup B$ , i.e.,  $T \cup C$  é consistente.

□

#### 4.2.2 – Forma\_Normal\_Conjuntiva\_Clausulas\_Horn(A, B, C)

O algoritmo `Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C)` (Figura 4.3) recebe conjuntos de proposições na Forma HF e converte-os para a FNC com cláusulas Horn.

```

Algoritmo Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C):
Entrada:
  • A, B e C, conjuntos proposições pertencentes à linhagem da LP
    na Forma HF.
Saída:
  • Conjunto R resultado da união e conversão dos conjuntos A, B e
    C para a FNC com cláusulas Horn, denotando  $r_1 \wedge r_2 \wedge r_3 \wedge \dots \wedge r_n$ .
1  {
2    S := A U B U C;
3    R := { };
4    para i := 1 até |S| faça
5      {
6        se S[i] é da Forma 1 então
7          R := R U  $a_1, a_2, a_3, \dots, a_n$ ;
8        se S[i] é da Forma 2 então
9          R := R U  $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n$ ;
10       se S[i] é da Forma 3 então
11         R := R U  $\neg a_1 \vee \neg a_2 \vee \neg a_3 \vee \dots \vee \neg a_n \vee b_1,$ 
12                  $\neg a_1 \vee \neg a_2 \vee \neg a_3 \vee \dots \vee \neg a_n \vee b_2,$ 
13                  $\neg a_1 \vee \neg a_2 \vee \neg a_3 \vee \dots \vee \neg a_n \vee b_3,$ 
14                  $\dots,$ 
15                  $\neg a_1 \vee \neg a_2 \vee \neg a_3 \vee \dots \vee \neg a_n \vee b_m$ ;
16       se S[i] é da Forma 4 então
17         R := R U  $a_1 \vee \neg b_1 \vee \neg b_2 \vee \neg b_3 \vee \dots \vee \neg b_m,$ 
18                  $a_2 \vee \neg b_1 \vee \neg b_2 \vee \neg b_3 \vee \dots \vee \neg b_m,$ 
19                  $a_3 \vee \neg b_1 \vee \neg b_2 \vee \neg b_3 \vee \dots \vee \neg b_m,$ 
20                  $\dots,$ 
21                  $a_n \vee \neg b_1 \vee \neg b_2 \vee \neg b_3 \vee \dots \vee \neg b_m$ ;
22       }
23     retornar R;
24   }

```

**Figura 4.3:** Algoritmo Forma\_Normal\_Conjuntiva\_Clausulas\_Horn(A, B, C).

Inicialmente, o algoritmo computa o conjunto S como a união dos conjuntos A, B e C (linha 2). O algoritmo então converte, uma a uma, as proposições da Forma HF, presentes em S para proposições na FNC com cláusulas Horn (linhas de 4 a 22).

**Exemplo 4.3.** Sejam  $T = \{ \text{interruptor\_desligado} \rightarrow \text{lampada\_apagada}, \text{bateria\_descarregada} \rightarrow \text{lampada\_apagada}, \text{fios\_rompidos} \rightarrow \text{lampada\_apagada} \}$ ,  $C = \{ \text{fios\_novos}, \text{fios\_novos} \rightarrow \neg \text{fios\_rompidos} \}$  e  $F = \{ \text{lampada\_apagada} \}$ . Suponhamos a seguinte chamada ao algoritmo: Forma\_Normal\_Conjuntiva\_Clausulas\_Horn(T, C,  $\neg F$ ). Nesta chamada T é passado para o parâmetro A, C é passado para o parâmetro B e  $\neg F$  é passado para o parâmetro C. A conversão para a FNC com cláusulas Horn realizada pelo algoritmo acontece a partir da geração do conjunto  $A \cup B \cup C$  (linha 2), o qual é armazenado na estrutura R (um

conjunto de cláusulas), e a conversão das proposições de acordo com a Forma HF que elas representam (linhas 4 a 22) ocorre assim:

- $S = \{ \text{interruptor\_desligado} \rightarrow \text{lampada\_apagada}, \text{bateria\_descarregada} \rightarrow \text{lampada\_apagada}, \text{fios\_rompidos} \rightarrow \text{lampada\_apagada}, \text{fios\_novos}, \text{fios\_novos} \rightarrow \neg \text{fios\_rompidos}, \neg \text{lampada\_apagada} \}$ .
- 1ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada} \}$ .
- 2ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada} \}$ .
- 3ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada} \}$ .
- 4ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \text{fios\_novos} \}$ .
- 5ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \text{fios\_novos}, \neg \text{fios\_novos} \vee \neg \text{fios\_rompidos} \}$ .
- 6ª iteração:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \text{fios\_novos}, \neg \text{fios\_novos} \vee \neg \text{fios\_rompidos}, \neg \text{lampada\_apagada} \}$ .

□

### 4.2.3 – Resolucao(R)

O algoritmo  $\text{Resolucao}(R)$  recebe como entrada o conjunto  $R$  de proposições na FNC com cláusulas Horn. O algoritmo então busca pelo primeiro átomo, reúne todas as cláusulas que possuem esse átomo, bem como todas as cláusulas que possuem a negação desse átomo. Após, utiliza a regra de Resolução da LP (Definição 2.17) em um produto cartesiano entre as cláusulas que contém o átomo e sua negação. Na Figura 4.4 encontra-se o pseudocódigo deste algoritmo.

```

Algoritmo Resolucao(R):
Entrada:
  • Um Conjunto finito  $R = \{ r_1, r_2, r_3, \dots, r_n \}$  de proposições na FNC com cláusulas Horn, denotando  $r_1 \wedge r_2 \wedge r_3 \wedge \dots \wedge r_n$ .
Saída:
  • Retorna o conjunto R após a Resolução.
1  {
2    enquanto há átomos em R faça
3    {
4      pa := Busca primeiro átomo em R;
5      Cp := Busca todas as cláusulas onde há pa;
6      Cn := Busca todas as cláusulas onde há  $\neg$ pa;
7      //As estruturas Cp e Cn são listas de cláusulas.
8      Retirar de R as cláusulas pertencentes a Cp e Cn;
9      para i := 1 até Tamanho(Cp) faça
10     {
11       para j := 1 até Tamanho(Cn) faça
12       {
13         R := R U resolvente(Cp[i], Cn[j], pa);
14       }
15     }
16   }
17   retornar R;
18 }

```

**Figura 4.4: Algoritmo Resolucao (R).**

**Exemplo 4.4.** Utilizando o conjunto  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, \neg lampada\_apagada \}$ , gerado no Exemplo 4.3, na aplicação do algoritmo Resolucao (R) temos:

- 1ª iteração:
  - A busca pelo primeiro átomo (linha 4):
    - $pa = lampada\_apagada$ .
  - A formação da lista Cp (linha 5):
    - $Cp = [ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada ]$ .
  - A formação da lista Cn (linha 6):
    - $Cn = [ \neg lampada\_apagada ]$ .
  - Na linha 8, a retirada dos elementos de Cp e Cn de R gera:
    - $R = \{ fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos \}$ .

- Nas linhas de 9 a 15 aplica-se a Regra de Resolução (em produto cartesiano entre os elementos de Cp e Cn):
  - A aplicação da resolução entre  $\neg interruptor\_desligado \vee lampada\_apagada$  e  $\neg lampada\_apagada$  deriva  $\neg interruptor\_desligado$ .
  - A aplicação da resolução entre  $\neg bateria\_descarregada \vee lampada\_apagada$  e  $\neg lampada\_apagada$  deriva  $\neg bateria\_descarregada$ .
  - A aplicação da resolução entre  $\neg fios\_rompidos \vee lampada\_apagada$  e  $\neg lampada\_apagada$  deriva  $\neg fios\_rompidos$ .
- A inserção do resultado das resoluções em R gera:
  - $R = \{ fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos \}$ .
- 2ª iteração:
  - A busca pelo primeiro átomo (linha 4):
    - $pa = fios\_novos$ .
  - A formação da lista Cp (linha 5):
    - $Cp = [ fios\_novos ]$ .
  - A formação da lista Cn (linha 6):
    - $Cn = [ \neg fios\_novos \vee \neg fios\_rompidos ]$ .
  - Na linha 8, a retirada dos elementos de Cp e Cn de R gera:
    - $R = \{ \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos \}$ .
  - Nas linhas de 9 a 15 aplica-se a Regra de Resolução (em produto cartesiano entre os elementos de Cp e Cn):
    - A aplicação da resolução entre  $fios\_novos$  e  $\neg fios\_novos \vee \neg fios\_rompidos$  deriva  $\neg fios\_rompidos$ .
  - A inserção do resultado das resoluções em R gera:
    - $R = \{ \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos, \neg fios\_rompidos \}$ .
  - Como o conjunto R é uma coleção distinta de elementos:

- $R = \{ \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos \}.$

□

#### 4.2.4 – Formular\_Hipoteses\_Candidatas(R)

O algoritmo `Formular_Hipoteses_Candidatas(R)` (Figura 4.5) recebe um conjunto de proposições resolvidas, i.e., após o cálculo feito pelo algoritmo `Resolucao(R)` e opera da seguinte maneira: (1) nega  $R$ , obtendo assim as  $m$  primeiras hipóteses candidatas  $h_1, h_2, h_3, \dots, h_m$ , (2) combina estas  $m$  primeiras hipóteses candidatas duas a duas, três a três, etc. até  $\lambda_2$  a  $\lambda_2$ , i.e., gera  $h_i \wedge h_j$  ( $i \neq j$ ) na primeira iteração,  $h_i \wedge h_j \wedge h_k$  ( $i \neq j \neq k$ ) na segunda iteração e  $h_i \wedge h_j \wedge \dots \wedge h_q$  ( $i \neq j \neq \dots \neq q$ ) na última iteração onde  $q$  é o mínimo entre  $m$  e  $\lambda_2$ .

```

Algoritmo Formular_Hipoteses_Cadidatas(R):
Entrada:
  • Um conjunto finito  $R = \{ r_1, r_2, r_3, \dots, r_n \}$ , de proposições
    pertencentes à linguagem da LP, denotando  $r_1 \wedge r_2 \wedge r_3 \wedge \dots \wedge r_n$ ,
    após o cálculo feito pelo algoritmo Resolucao(R).
Saída:
  • Hipóteses  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , um conjunto finito de
    hipóteses candidatas, denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ .
1  {
2    S := { };
3    para i := 1 até |R| faça
4      S := S U  $\neg R[i]$ ;
5    qtdCombinacoes := Min(|S|,  $\lambda_2$ );
6    H := S;
7    para i := 2 até qtdCombinacoes faça
8      {
9        H := H U Combinação dos elementos de S, i a i;
10       //A combinação consiste em combinar os elementos de
           S, 2 a 2, 3 a 3, 4 a 4, ..., i a i. Os elementos
           resultantes da combinação são conectados com o
           conectivo  $\wedge$ .
11      }
12    retornar H;
13  }

```

**Figura 4.5: Algoritmo Formular\_Hipoteses\_Candidatas(R)**

Por meio da combinação das hipóteses pode-se formular hipóteses com força explanatória maior, e.g., para uma hipótese  $p$  que explica um fato  $x$  e uma hipótese  $q$  que explica um fato  $y$  e para um conjunto de fatos observados tem-se  $\{ x, y \}$ , ambas as

hipóteses  $p$  e  $q$  possuem força explanatória igual a 0,5, no entanto uma combinação de ambas, portanto  $p \wedge q$ , possui força explanatória igual 1.

**Exemplo 4.5.** Considere que o algoritmo recebe como entrada  $R = \{ \neg interruptor\_desligado, \neg bateria\_descarregada, \neg fios\_rompidos \}$ . O algoritmo nega cada elemento do conjunto  $R$  para obter as primeiras hipóteses  $S = \{ interruptor\_desligado, bateria\_descarregada, fios\_rompidos \}$  (linhas 3 e 4). Em seguida calcula a quantidade máxima de combinações. Esta quantidade é o mínimo entre a quantidade de elementos de  $S$  e o valor de  $\lambda_2$  (valor da complexidade máxima, linha 5). Assumindo  $\lambda_2 = 5$  então a quantidade de combinações  $qtdCombinacoes$  será igual a 3 (linha 5). Após isto, os elementos de  $S$  são combinados 2 a 2, 3 a 3 e conectados pelo conectivo  $\wedge$  e inseridos em  $H$  (linhas 7 a 11).

**Combinação 2 a 2:**

- $H = \{ interruptor\_desligado, bateria\_descarregada, fios\_rompidos, interruptor\_desligado \wedge bateria\_descarregada, interruptor\_desligado \wedge fios\_rompidos, bateria\_descarregada \wedge fios\_rompidos \}$

**Combinação 3 a 3:**

- $H = \{ interruptor\_desligado, bateria\_descarregada, fios\_rompidos, interruptor\_desligado \wedge bateria\_descarregada, interruptor\_desligado \wedge fios\_rompidos, bateria\_descarregada \wedge fios\_rompidos, interruptor\_desligado \wedge bateria\_descarregada \wedge fios\_rompidos \}$ .

□

**4.2.5 – Remover\_Hipoteses\_Inconsistentes(T, C, H)**

O algoritmo `Remover_Hipoteses_Inconsistentes(T, C, H)` (Figura 4.6) opera da seguinte maneira. Para cada  $h \in H$ : (1) o algoritmo converte as proposições do conjunto  $T \cup C \cup \{ h \}$  para a FNC com cláusulas Horn, (2) aplica a resolução ao sistema de proposições resultante e (3) se uma cláusula vazia está presente no resultado da resolução a hipótese  $h$  é retirada de  $H$ .

```

Algoritmo Remover_Hipoteses_Inconsistentes(T, C, H);
Entrada:
  • Um conjunto finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ .
  • Um conjunto finito,  $C = \{ c_1, c_2, c_3, \dots, c_p \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_p$ .
  • Um conjunto finito,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP, denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto representa as hipóteses candidatas.
Saída:
  • Um conjunto finito,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP, denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto representa as hipóteses consistentes com T e C.
1  {
2     $H' := H$ ;
3    para  $i := 1$  até  $|H'|$  faça
4      {
5         $R := \text{Forma\_Normal\_Conjuntiva\_Clausulas\_Horn}(T, C,$ 
            $H'[i])$ ;
6         $R := \text{Resolucao}(R)$ ;
7        se existir alguma cláusula vazia em R então
8           $H := H - \{ H'[i] \}$ ;
9      }
10   retornar H;
11 }

```

**Figura 4.6: Algoritmo Remover\_Hipoteses\_Inconsistentes(T, C, H).**

**Exemplo 4.6.** Como exemplo de funcionamento do algoritmo `Remover_Hipoteses_Inconsistentes(T, C, H)`, considere  $T = \{ \text{interruptor\_desligado} \rightarrow \text{lampada\_apagada}, \text{bateria\_descarregada} \rightarrow \text{lampada\_apagada}, \text{fios\_rompidos} \rightarrow \text{lampada\_apagada} \}$ ,  $C = \{ \text{fios\_novos}, \text{fios\_novos} \rightarrow \neg \text{fios\_rompidos} \}$  e  $H = \{ \text{interruptor\_desligado}, \text{bateria\_descarregada}, \text{fios\_rompidos}, \text{interruptor\_desligado} \wedge \text{bateria\_descarregada}, \text{interruptor\_desligado} \wedge \text{fios\_rompidos}, \text{bateria\_descarregada} \wedge \text{fios\_rompidos}, \text{interruptor\_desligado} \wedge \text{bateria\_descarregada} \wedge \text{fios\_rompidos} \}$ . Inicialmente o algoritmo produz  $H'$  igual ao conjunto  $H$  recebido como entrada. Para cada proposição  $h$  em  $H'$ , o algoritmo converte  $T \cup C \cup \{ h \}$  para a FNC com cláusulas Horn (linha 5), armazenando o resultado em  $R$ , e aplica algoritmo de resolução sobre  $R$  (linha 6). Se existir alguma cláusula vazia em  $R$  então  $\{ h \}$  é retirada do conjunto  $H$ . Para os conjuntos  $T, C$  e  $H$  deste exemplo temos:

- 1ª iteração ( $i = 1, H'[1] = \text{interruptor\_desligado}$ ):

- Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, interruptor\_desligado \}$
- Na linha 6, após a resolução:  $R = \{ lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, \neg fios\_rompidos \}$
- A aplicação da resolução não gerou cláusula vazia (teste linha 7), portanto a hipótese *interruptor\_desligado* é consistente, permanecendo em H.
- 2ª iteração ( $i = 2, H'[2] = bateria\_descarregada$ ):
  - Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, bateria\_descarregada \}$
  - Na linha 6, após a resolução:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, \neg fios\_rompidos \}$
  - A aplicação da resolução não gerou cláusula vazia (teste linha 7), portanto a hipótese *bateria\_descarregada* é consistente, permanecendo em H.
- 3ª iteração ( $i = 3, H'[3] = fios\_rompidos$ ):
  - Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, fios\_rompidos \}$
  - Na linha 6, após a resolução:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, \{ \} \}$
  - A aplicação da resolução gerou uma cláusula vazia (teste linha 7, portanto a hipótese *fios\_rompidos* é inconsistente.
  - Na linha 8 retira-se *fios\_rompidos* de H.

- 4ª iteração ( $i = 4$ ,  $H'[4] = interruptor\_desligado \wedge bateria\_descarregada$ ):
  - Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, interruptor\_desligado, bateria\_descarregada \}$
  - Na linha 6, após a resolução:  $R = \{ lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, \neg fios\_rompidos \}$
  - A aplicação da resolução não gerou cláusula vazia (teste linha 7), portanto a hipótese  $interruptor\_desligado \wedge bateria\_descarregada$  é consistente, permanecendo em H.
- 5ª iteração ( $i = 5$ ,  $H'[5] = interruptor\_desligado \wedge fios\_rompidos$ ):
  - Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, interruptor\_desligado, fios\_rompidos \}$
  - Na linha 6, após a resolução:  $R = \{ lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \{ \} \}$
  - A aplicação da resolução gerou uma cláusula vazia (teste linha 7, portanto a hipótese  $interruptor\_desligado \wedge fios\_rompidos$  é inconsistente.
  - Na linha 8 retira-se  $interruptor\_desligado \wedge fios\_rompidos$  de H.
- 6ª iteração ( $i = 6$ ,  $H'[6] = bateria\_descarregada \wedge fios\_rompidos$ ):
  - Na linha 5:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \neg bateria\_descarregada \vee lampada\_apagada, \neg fios\_rompidos \vee lampada\_apagada, fios\_novos, \neg fios\_novos \vee \neg fios\_rompidos, bateria\_descarregada, fios\_rompidos \}$
  - Na linha 6, após a resolução:  $R = \{ \neg interruptor\_desligado \vee lampada\_apagada, \{ \} \}$
  - A aplicação da resolução gerou uma cláusula vazia (teste linha 7, portanto a hipótese  $bateria\_descarregada \wedge fios\_rompidos$  é inconsistente.
  - Na linha 8 retira-se  $bateria\_descarregada \wedge fios\_rompidos$  de H.
- 7ª iteração ( $i = 7$ ,  $H'[7] = interruptor\_desligado \wedge bateria\_descarregada \wedge fios\_rompidos$ ):

- Na linha 5:  $R = \{ \neg \text{interruptor\_desligado} \vee \text{lampada\_apagada}, \neg \text{bateria\_descarregada} \vee \text{lampada\_apagada}, \neg \text{fios\_rompidos} \vee \text{lampada\_apagada}, \text{fios\_novos}, \neg \text{fios\_novos} \vee \neg \text{fios\_rompidos}, \text{interruptor\_desligado}, \text{bateria\_descarregada}, \text{fios\_rompidos} \}$
- Na linha 6, após a resolução:  $R = \{ \text{lampada\_apagada}, \{ \} \}$
- A aplicação da resolução gerou uma cláusula vazia (teste linha 7, portanto a hipótese  $\text{interruptor\_desligado} \wedge \text{bateria\_descarregada} \wedge \text{fios\_rompidos}$  é inconsistente.
- Na linha 8 retira-se  $\text{interruptor\_desligado} \wedge \text{bateria\_descarregada} \wedge \text{fios\_rompidos}$  de H.

O algoritmo tem como resposta, portanto,  $H = \{ \text{interruptor\_desligado}, \text{bateria\_descarregada}, \text{interruptor\_desligado} \wedge \text{bateria\_descarregada} \}$

□

#### 4.2.6 – Selecionar\_Boas\_Hipoteses(T, C, H, F)

O algoritmo `Selecionar_Boas_Hipoteses(T, C, H, F)` (Figura 4.7) opera da seguinte maneira: (1) tendo como entrada T, C, H e F computa a força explanatória e a complexidade de cada hipótese  $h \in H$ , (2) remove as hipóteses  $h$  cuja força explanatória seja menor que a constante  $\lambda_1$ , (3) computa o conjunto E que contem aquelas hipóteses de H com maior força explanatória, (4) contendo as hipóteses de E com a menor complexidade e; (5) retorna X como resposta.

**Algoritmo** Selecionar\_Boas\_Hipoteses(T, C, H, F):

**Entrada:**

- Um conjunto finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ .
- Um conjunto finito,  $C = \{ c_1, c_2, c_3, \dots, c_p \}$ , de proposições pertencentes à linguagem da LP, na Forma HF, denotando  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_p$ .
- Um conjunto finito,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , de proposições pertencentes à linguagem da LP, denotando  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . Este conjunto deve conter hipóteses consistentes com T e C.
- Um conjunto finito e não vazio de fatos,  $F = \{ f_1, f_2, f_3, \dots, f_q \}$ , de átomos pertencentes à linguagem da LP, denotando  $f_1 \wedge f_2 \wedge f_3 \wedge \dots \wedge f_q$ .

**Saída:**

- Um conjunto finito,  $X = \{ x_1, x_2, x_3, \dots, x_n \}$ , de proposições pertencentes à linguagem da LP, denotando  $x_1 \vee x_2 \vee x_3 \vee \dots \vee x_n$ . Este conjunto representa boas hipóteses que explicam os fatos em F.

```
1  {
2    //Computa a força explanatória de cada hipótese  $h \in H$ 
      e armazena o resultado no vetor FE.
3    para i := 1 até |H| faça
4      FE[i] := 0;
5      S := T U C;
6      para i := 1 até |H| faça
7        {
8          para j := 1 até |F| faça
9            {
10             R := Forma_Normal_Conjuntiva_Clausulas_Horn(S,
11               H[i],  $\neg F[j]$ );
12             R := Resolucao(R);
13             se há cláusulas vazias em R então
14               FE[i] := FE[i] + 1;
15           }
16         }
17       H' := H;
18       //Remove as hipóteses h cuja força explanatória seja
19       menor que a constante  $\lambda_1$ .
20       para i := 1 até |H'| faça
21         se FE[i] <  $\lambda_1$  então
22           H := H - H'[i];
23       //Computa o conjunto E de hipóteses contendo a maior
24       força explanatória em H.
25       m := Maior(FE);
26       E := { };
27       para i := 1 até |H| faça
28         se FE[i] = m então
29           E := E U H[i];
30       //Computa o conjunto X de hipóteses contendo a menor
31       complexidade em E.
32       m := Complexidade da hipótese com menor complexidade de
33       E;
34       X := { };
35       para i := 1 até |E| faça
36         se Complexidade(E[i]) = m então
37           X := X U E[i];
38       retornar X;
39     }
```

**Figura 4.7: Algoritmo Selecionar\_Boas\_Hipoteses(T, C, H, F).**

**Exemplo 4.7.** Como exemplo do funcionamento do algoritmo `Selecionar_Boas_Hipotese`( $T, C, H, F$ ),  $T = \{ \textit{interruptor\_desligado} \rightarrow \textit{lampada\_apagada}, \textit{bateria\_descarregada} \rightarrow \textit{lampada\_apagada}, \textit{fios\_rompidos} \rightarrow \textit{lampada\_apagada} \}$ ,  $C = \{ \textit{fios\_novos}, \textit{fios\_novos} \rightarrow \neg \textit{fios\_rompidos} \}$  e  $H = \{ \textit{interruptor\_desligado}, \textit{bateria\_descarregada}, \textit{interruptor\_desligado} \wedge \textit{bateria\_descarregada} \}$  e  $F = \{ \textit{lampada\_apagada} \}$ . O algoritmo calcula a força explanatória de cada hipótese em  $H$  (linhas 6 à 15). Retira as hipóteses de  $H$  que tenham a força explanatória menor que  $\lambda_1$  (linhas 18 à 20). Coleta as hipóteses de maior força explanatória e menor complexidade (linhas 22 à 32). Para os conjuntos deste exemplo:

- Após a execução das linhas 6 à 15 se obtém:  $FE[1] = 1$  indicando que a força explanatória da hipótese *interruptor\_desligado* é igual a 1, pois  $T \cup C \cup \{ \textit{interruptor\_desligado} \} \models \{ \textit{lampada\_apagada} \}$ ;  $FE[2] = 1$  indicando que a força explanatória da hipótese *bateria\_descarregada* é igual a 1, pois  $T \cup C \cup \{ \textit{bateria\_descarregada} \} \models \{ \textit{lampada\_apagada} \}$  e;  $FE[3] = 1$  indicando que a força explanatória da hipótese *interruptor\_desligado*  $\wedge$  *bateria\_descarregada* é igual a 1, pois  $T \cup C \cup \{ \textit{interruptor\_desligado} \wedge \textit{bateria\_descarregada} \} \models \{ \textit{lampada\_apagada} \}$ ;
- Neste exemplo, na execução das linhas 18 à 20 não é retirada nenhuma hipótese, pois todas as hipóteses possuem força explanatória igual à 1 o que é maior que  $\lambda_1 = 0,5$ .
- Para a seleção das hipóteses com maior força explanatória (linhas 22 à 26), temos a variável  $m = 1$ , representando a maior força explanatória dentre todas as hipóteses, e as hipóteses selecionadas para fazer parte do conjunto  $E$  são *interruptor\_desligado*, *bateria\_descarregada* e *interruptor\_desligado*  $\wedge$  *bateria\_descarregada*, devido as suas forças explanatórias serem maiores ou iguais à  $m$ .
- Para a seleção das hipóteses com menor complexidade (linhas 28 à 32, temos a variável  $m = 1$ , representando a menor complexidade dentre todas as hipóteses, e as hipóteses selecionadas para fazer parte do conjunto  $X$  são *interruptor\_desligado* e *bateria\_descarregada*, pois suas complexidades são iguais a  $m$ , *interruptor\_desligado*  $\wedge$  *bateria\_descarregada* é descartada, pois complexidade é maior que  $m$ .

- Por fim o resultado do algoritmo é o conjunto  $X = \{ \text{interruptor\_desligado}, \text{bateria\_descarregada} \}$ .

□

### 4.3 – Tempo de execução

Na análise do tempo de execução do algoritmo Peirce, começaremos pelo algoritmo `Resolucao(R)` uma vez que ele compõe a maioria dos algoritmos.

Considerando os tamanhos de `R` (linha 2), `Cp` (linha 5) e `Cn` (linha 6), o tempo de execução deste algoritmo é  $O(n^3)$ . Este cálculo serve aos nossos propósitos, mas cabe observar que está superestimado, pois não é possível que o tamanho igual a  $n$ , que é a quantidade de átomos, ocorra simultaneamente em `R`, `Cp` e `Cn`.

O algoritmo de conversão `Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C)` possui tempo de execução  $O(n)$ , uma vez que se todas as proposições encontram-se na Forma HF, são diretamente convertidas em cláusulas em  $O(1)$ .

Uma vez que o algoritmo `Resolucao(R)` possui tempo de execução  $O(n^3)$ , o algoritmo `Consistentes(A, B)`, também possui tempo de execução  $O(n^3)$ , uma vez que é composto por `Forma_Normal_Conjuntiva_Clausulas_Horn(A, B, C)`, cujo tempo de execução é  $O(n)$  e por `Resolucao(R)`, cujo tempo de execução é  $O(n^3)$ .

O algoritmo `Formular_Hipoteses_Candidatas(R)` possui tempo de execução  $O(n^{\lambda_2})$ , pois produz no máximo  $O(n^2) + O(n^3) + \dots + O(n^{\lambda_2})$  combinações de hipóteses.

O algoritmo `Remove_Hipoteses_Inconsistentes(T, C, H)` possui tempo de execução  $O(n^{3+\lambda_2})$  por realizar no máximo uma quantidade constante de  $O(n^{\lambda_2})$  resoluções de tempo de execução  $O(n^3)$ .

Na seleção de boas hipóteses o algoritmo `Selecionar_Boas_Hipoteses(T, C, H, F)` utiliza-se resolução  $O(n^3)$  para se computar a força explanatória de  $n$  hipóteses, sendo assim seu tempo de execução é de  $O(n^4)$  no máximo.

Ao todo o algoritmo `Peirce(T, C, F)` possui tempo de execução  $O(n^{3+\lambda_2})$ . Como  $\lambda_2$  é uma constante, tipicamente igual a 5, o algoritmo possui tempo de execução polinomial.

## 4.4 – Pesquisa Experimental

Nesta seção é descrita uma Pesquisa Experimental com pessoas. Esta pesquisa foi realizada com o objetivo verificar se as hipóteses computadas na solução do algoritmo Peirce coincidem com hipóteses formuladas por pessoas quando realizam raciocínios abduativos. A motivação desta pesquisa parte do entendimento deste trabalho de que a seleção de boas hipóteses abduativas e a estrutura TCHF propostas são conceitos inspirados por fatores humanos.

### 4.4.1 – Hipóteses de pesquisa

A Pesquisa Experimental verifica a tendência das hipóteses computadas pelo algoritmo Peirce coincidirem com raciocínios abduativos realizados por pessoas, e pode ser assim descrita, segundo a tradição Estatística de declaração de hipóteses nulas e alternativas:

- $H_0$ : Pessoas *não formulam hipóteses coincidentes com aquelas formuladas pelo algoritmo Peirce* como solução de problemas abduativos.
- $H_a$ : Pessoas *formulam hipóteses coincidentes com aquelas formuladas pelo algoritmo Peirce* como solução de problemas abduativos.

### 4.4.2 – Materiais

Com o intuito de verificar as hipóteses levantadas na Seção 4.4.1, foi elaborada uma pesquisa por meio de um Questionário. O Questionário foi validado por um teste piloto que envolveu 25 estudantes de graduação os quais não participaram do teste final. O questionário está reproduzido integralmente no Apêndice III.

No Questionário são apresentadas situações, onde se faz necessário a realização do raciocínio abduativo. Cada questão apresenta uma situação onde é descrito, em português corrente e de maneira implícita, uma teoria, fatos observados e, em alguns casos, condições aceitas. Nas alternativas de cada questão apresentam-se possíveis hipóteses abduativas.

O Questionário é composto por 10 questões envolvendo situações nas quais se quer verificar se a solução será pautada pela escolha de alternativas que corresponde às

hipóteses computadas pelo algoritmo Peirce. A ordem em que foram dispostas obedeceu a uma função referente ao agrupamento entre cenários semelhantes (tempo, diagnóstico médico, investigação policial etc.).

#### 4.4.3 – Sujeitos da pesquisa

Um total de 193 estudantes de graduação e pós-graduação da Faculdade Campo Limpo Paulista (FACCAMP), participaram da pesquisa. Os cursos/turmas dos alunos que participaram da pesquisa foram aleatoriamente estabelecidos. A Tabela 4.1 sintetiza o perfil dos participantes.

**Tabela 4.1: Perfil dos participantes da pesquisa.**

<i>Sexo, n (%)</i>			<i>Idade, em anos</i>		
<i>Masculino</i>	<i>Feminino</i>	<i>Não Identificados</i>	<i>Faixa</i>	<i>Média</i>	<i>Mediana</i>
81 (42%)	102 (53%)	10 (5%)	18 – 60	27	24

De forma geral, pode-se dizer que o perfil dos participantes apresenta um pequeno predomínio de pessoas do sexo feminino, da faixa etária de 18 a 60 anos de idade, mas com média e mediana próximas de 25 anos, ou seja, mais próximas do limite inferior da faixa etária.

#### 4.4.4 – Procedimentos experimentais

O Questionário da pesquisa, impresso em papel, foi desenvolvido para ser respondido em 15 minutos, tempo adequado segundo Mackenzie (2013) para manter os participantes focados e evitar a fadiga. O experimento foi constituído por diversas Sessões Experimentais, cada uma delas realizada com um curso/turma de alunos. Cada Sessão foi conduzida em sala de aula no momento inicial de uma aula, durante a última semana do mês de novembro de 2013 e foi constituída pelas seguintes fases:

1. O experimentador descreveu aos participantes o propósito do estudo e solicitou a participação voluntária de pessoas;
2. O experimentador fez um pequeno treinamento dos participantes por meio da leitura e análise da Situação Exemplo que consta da primeira página do

Questionário (Apêndice III). O experimentador enfatizou que, para cada questão, o participante deveria escolher a alternativa que o deixava mais convicto de que a sua escolha era uma boa hipótese para explicar a situação apresentada;

3. Os participantes responderam ao Questionário da pesquisa;
4. O experimentador finalizou agradecendo as participações voluntárias e explicou a utilidade dos resultados da pesquisa.

As respostas ao Questionário foram digitadas, sendo atribuído 1 ponto para cada questão respondida coincidentemente com a hipótese computada pelo algoritmo Peirce.

Posteriormente, eliminou-se respostas incompletas e de participantes desatentos ou não zelosos. O critério utilizado para a eliminação foi a divergência de acertos em relação à média de todos os participantes. Foram eliminadas respostas que apresentaram quantidade de acertos, menores do que a média, com uma probabilidade de terem sido produzidas ao acaso maior ou igual a 99%. Ao final do processo de eliminação a quantidade respostas válidas ficou igual a 133.

#### 4.4.5 – Resultados e discussão

Como medida estatística da significância da aderência dos participantes às alternativas que sinalizam escolhas coincidentes às hipóteses formuladas pelo algoritmo Peirce, utilizou-se o teste Qui-Quadrado ( $\chi^2$ ) (Callegari-Jacques, 2008), ao nível de significância de 1%. A Tabela 4.2 apresenta o resultado da verificação da tendência das soluções computadas pelo algoritmo Peirce coincidirem com as soluções formuladas por pessoas.

**Tabela 4.2: Resultado da verificação da tendência das soluções computadas pelo algoritmo Peirce coincidirem com as soluções formuladas por pessoas.**

<i>Respostas aderentes</i>	<i>Respostas não aderentes</i>	$\chi^2$	<i>Valor p</i>
86	47	11,44	0,001

Para a população estudada o teste  $\chi^2$  sugere que a coincidência entre as respostas dos participantes e as soluções computadas pelo algoritmo Peirce é significativa ( $\chi^2 (1) = 11,44$ ,  $p = 0,001 < 0,01$ ). Ou seja, admite-se a hipóteses alternativa formulada na Seção 4.4.1, segundo a qual pessoas formulam hipóteses coincidentes com aquelas formuladas pelo algoritmo Peirce como solução de problemas abduativos.

Por fim, é importante ressaltar, mais uma vez, que este experimento não esgota o assunto, mas busca apontar um caminho para a compreensão do fenômeno da formulação e seleção de boas hipóteses como um fator humano.

# Capítulo 5

## Peirce Online

*Este capítulo apresenta o Sistema Peirce Online para a realização de raciocínios abduativos. A Seção 5.1 introduz o Sistema Peirce Online com exemplos de utilização. A Seção 5.2 apresenta a arquitetura cliente-servidor por trás do Sistema. A Seção 5.3 apresenta a gramática utilizada para gerar proposições na linguagem Peirce Abduativa. E a Seção 5.4 descreve o Compilador para a linguagem Peirce Abduativa.*

## 5.1 – Peirce Online

O Sistema Peirce Online<sup>3</sup> é um protótipo desenvolvido no âmbito desta dissertação de mestrado; ele realiza raciocínios abduativos contemplando a proposta apresentada nos Capítulos 3 e 4, i.e., o Sistema realiza raciocínios abduativos utilizando a estrutura quaternária de raciocínio TCHF e utiliza um critério para a seleção de boas hipóteses que consiste na força explanatória e complexidade de cada hipótese.

Peirce Online é implementado em um ambiente Web utilizando HTML e ASP.NET, juntamente com a linguagem de programação C#.

Todas as páginas do sistema apresentam um cabeçalho cujos menus são *links* para acessar outras páginas, do sistema e externas, incluindo a página principal, onde os usuários podem realizar raciocínios abduativos, (*menu: Peirce*), um tutorial (*menu: How to use Peirce*), um redirecionamento para a página do Programa de Mestrado em Ciência da Computação da FACCAMP (*menu: Faccamp*) e uma página descrevendo a versão, a data de desenvolvimento e a relação dos desenvolvedores (*menu: About*). Especialmente para assistir a banca na avaliação desta dissertação, a página *How to use Peirce* contém um link para uma página onde podem ser obtidos o texto (fonte) dos exemplos deste e de outros Capítulos.

A Figura 5.1 apresenta a página onde os usuários podem interagir com o Sistema e realizar raciocínios abduativos. Esta página é composta pelos botões *New*, *Open*, *Save* (no início da página) e *Abductive Reasoning* (mais próximo do fim da página). Os campos *Minimum Explanatory Power* ( $\lambda_1$ ) e *Maximum Complexity* ( $\lambda_2$ ) representam, respectivamente, valores para a força explanatória mínima e a complexidade máxima (Capítulo 3, Seção 3.3) que Peirce Online utilizará na realização do raciocínio abduativo. A página apresenta também os campos para descrição dos conjuntos Teoria, Condições Aceitas e Fatos, respectivamente, *Theory set*, *Conditions set* e *Facts set*. Acima dos campos *Theory set* e *Conditions set* existem botões para facilitar a inserção de símbolos especiais para conectivos lógicos, símbolos estes frequentemente não disponíveis para edição diretamente por meio de teclados convencionais. Ao fim da página, o campo *Hypotheses set* está presente para descrever o conjunto de hipóteses resultantes do raciocínio abduativo realizado pelo Sistema Peirce Online.

---

<sup>3</sup> <http://peirceonline.gear.host/>

# Peirce Online

Peirce

How to use Peirce

Faccamp

About

New Open Save

Minimum Explanatory Power ( $\lambda_1$ ):

Maximum Complexity ( $\lambda_2$ ):

Theory set:

```
{  
}
```

Conditions set:

```
{  
}
```

Facts set:

```
{  
}
```

Abductive Reasoning

Hypotheses set:

```
{  
}
```

**Figura 5.1: Página do Sistema Peirce Online. O Sistema permite realizar raciocínios abduativos declarando-se conjuntos Teoria, Condições Aceitas e Fatos.**

Peirce Online opera com conjuntos de proposições pertencentes à linguagem da LP. Para a definição de átomos estes devem iniciar com letras e podem ser seguidos por letras, números ou o símbolo subtração (“\_”), exceto para as palavras reservadas que representam conectivos lógicos (Tabela 5.1). Os conjuntos de proposições estão restritos à Forma HF, apresentada no Capítulo 2 (Definição 2.16), e são declarados entre chaves (“{}”); cada proposição pertencente aos conjuntos deve ser separada por vírgula (“,”). A Figura 5.2 rerepresenta, para facilidade de leitura deste texto, a Forma HF para a escrita de proposições.

**Forma HF:** Uma proposição pertencente a linguagem da LP na Forma HF é uma proposição escrita nos seguintes formatos:

- **Forma 1:**  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são literais.
- **Forma 2:**  $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são átomos negados.
- **Forma 3:**  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n \rightarrow b_1 \wedge b_2 \wedge b_3 \wedge \dots \wedge b_m$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são átomos e  $b_j$  ( $1 \leq j \leq m$ ) são literais.
- **Forma 4:**  $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n \rightarrow b_1 \vee b_2 \vee b_3 \vee \dots \vee b_m$ , em que  $a_i$  ( $1 \leq i \leq n$ ) são literais e  $b_j$  ( $1 \leq j \leq m$ ) são átomos negados.

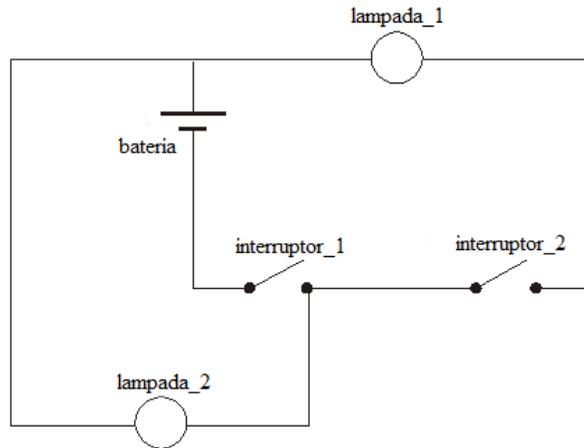
**Figura 5.2: Forma HF para ser utilizada nas proposições do Sistema Peirce Online.**

Os campos *Theory set* e *Conditions set* devem ser preenchidos com um conjunto de proposições podendo ser utilizadas todas as formas pertencentes à Forma HF. O campo *Facts set* deve ser preenchido com um conjunto de átomos separados por vírgula. O Exemplo 5.1 apresenta uma utilização do Sistema Peirce Online para criar um modelo para um circuito elétrico e, assim, permitir a realização de raciocínios abduativos sobre o modelo estabelecido.

**Exemplo 5.1.** A Figura 5.3 apresenta um circuito elétrico simples composto de duas lâmpadas (*lampada\_1* e *lampada\_2*), uma bateria (*bateria*), dois interruptores (*interruptor\_1* e *interruptor\_2*) e fios. Neste exemplo, propomos modelar uma teoria sobre explicações que poderiam levar as lâmpadas do circuito estarem apagadas. As proposições atômicas utilizadas no modelo são as seguintes:

- *bateria\_sem\_carga*: para nomear o fato de que a bateria não tem carga.
- *lampada\_1\_apagada*: para nomear o fato da lâmpada 1 estar apagada.
- *lampada\_2\_apagada*: para nomear o fato da lâmpada 2 está apagada.
- *interruptor\_1\_desligado*: para nomear o fato do interruptor 1 estar desligado.
- *interruptor\_2\_desligado*: para nomear o fato do interruptor 1 estar desligado.
- *fios\_rompidos*: para nomear o fato de fios estarem rompidos.

As proposições não necessitam ser declaradas em uma área especialmente destinada a isto, como ocorrem com variáveis em linguagens de programação como Pascal. Basta nomeá-las em algum campo. Elas só estão descritas aqui para melhor entendimento do leitor.



**Figura 5.3:** Circuito elétrico simples, composto de duas lâmpadas, uma bateria, dois interruptores e fios.

Para este exemplo, os campos do Peirce Online devem ser preenchidos como na Figura 5.4-a-b-c.

**Theory set:**

```
{
// Teoria sobre explicações que poderiam levar as lâmpadas do circuito
// estarem apagadas.
bateria_sem_carga → lampada_1_apagada ∧ lampada_2_apagada,
fios_rompidos → lampada_1_apagada ∧ lampada_2_apagada,
interruptor_1_desligado → lampada_1_apagada ∧ lampada_2_apagada,
interruptor_2_desligado → lampada_1_apagada
}
```

(a)

**Conditions set:**

```
{
}
```

(b)

**Facts set:**

```
{
lampada_1_apagada
}
```

(c)

**Hypotheses set:**

```
{
  bateria_sem_carga, fios_rompidos,
  interruptor_1_desligado,
  interruptor_2_desligado
}
```

(d)

**Figura 5.4: Exemplo da utilização do Sistema Peirce Online para criar um modelo de um circuito elétrico e permitir a realização de raciocínios sobre este modelo. Os campos *Theory set* e *Conditions set*, em conjunto, estabelecem o modelo. O campo *Facts set* permite descrever fatos. O campo *Hypotheses set* descreve o raciocínio abduutivo realizado pelo Sistema, considerando os fatos e modelo apresentados.**

Para execução do raciocínio abduutivo deve-se clicar no botão *Abductive Reasoning*. A solução do raciocínio é exibida no campo *Hypotheses set*, como exemplifica a Figura 5.4-d.

□

O Exemplo 5.2 amplia o Exemplo 5.1 acrescentando uma teoria sobre explicações para as lâmpadas estarem acesas. Este exemplo apresenta também uma importante utilização do campo *Conditions Set*. Proposições neste campo descreverão correlações lógicas para as duas teorias funcionarem conjuntamente.

**Exemplo 5.2.** Neste exemplo propomos ampliar a teoria acrescentando explicações que poderiam levar as lâmpadas do circuito a estarem acesas. Assim novas proposições atômicas são definidas:

- *bateria\_com\_carga*: para nomear o fato de que a bateria tem carga.
- *lampada\_1\_acesa*: para nomear o fato da lâmpada 1 estar acesa.
- *lampada\_2\_acesa*: para nomear o fato da lâmpada 2 estar acesa.
- *interruptor\_1\_ligado*: para nomear o fato do interruptor 1 estar ligado.
- *interruptor\_2\_ligado*: para nomear o fato do interruptor 2 estar ligado.
- *fios\_conectados*: para nomear o fato de fios estarem conectados.

Para o modelo deste exemplo, os campos do Peirce Online devem ser preenchidos como na Figura 5.5-a-b-c.

**Theory set:**

```
{
  // Teoria 1: Teoria sobre explicações que poderiam levar as lâmpadas do
  // circuito estarem apagadas.
  bateria_sem_carga → lampada_1_apagada ∧ lampada_2_apagada,
  fios_rompidos → lampada_1_apagada ∧ lampada_2_apagada,
  interruptor_1_desligado → lampada_1_apagada ∧ lampada_2_apagada,
  interruptor_2_desligado → lampada_1_apagada,
```

```
// Teoria 2: Teoria sobre explicações que poderiam levar as lâmpadas do circuito
//estarem acesas.
bateria_com_carga  $\wedge$  fios_conectados  $\wedge$  interruptor_1_ligado  $\rightarrow$  lampada_2_acesa,
bateria_com_carga  $\wedge$  fios_conectados  $\wedge$  interruptor_1_ligado  $\wedge$ 
interruptor_2_ligado  $\rightarrow$  lampada_1_acesa  $\wedge$  lampada_2_acesa
}
```

(a)

**Conditions set:**

```
{
// Proposições descrevendo correlações lógicas entre
// proposições da Teoria 1 e da Teoria 2.
bateria_com_carga  $\rightarrow$   $\neg$ bateria_sem_carga,
interruptor_1_desligado  $\rightarrow$   $\neg$ interruptor_1_ligado,
interruptor_2_desligado  $\rightarrow$   $\neg$ interruptor_2_ligado,
fios_rompidos  $\rightarrow$   $\neg$ fios_conectados
}
```

(b)

**Facts set:**

```
{
lampada_1_apagada, lampada_2_acesa
}
```

(c)

**Hypotheses set:**

```
{
bateria_com_carga  $\wedge$  fios_conectados  $\wedge$  interruptor_1_ligado  $\wedge$ 
interruptor_2_desligado
}
```

(d)

**Figura 5.5: Outro exemplo de utilização do Sistema Peirce Online.** Neste exemplo o Sistema é usado para criar um modelo para um circuito elétrico por meio de duas teorias, uma contendo proposições para o acendimento de lâmpadas e a outra teoria, proposições para o não acendimento de lâmpadas.

O campo *Conditions set* é utilizado para correlacionar as proposições das duas teorias. O campo *Hypotheses set* descreve o raciocínio abduutivo realizado pelo sistema, considerando os fatos e modelo apresentados.

Ao clicar no botão *Abductive Reasoning* a solução do raciocínio é exibida no campo *Hypotheses set*, como exemplifica a Figura 5.5-d.

□

Erros podem ocorrer durante a utilização do sistema Peirce Online. Estes erros podem ter relação com o preenchimento dos campos ou à realização do raciocínio abduutivo. Caso algum erro venha a ocorrer o Sistema exibe uma mensagem contendo uma descrição do erro no campo *Hypotheses set*.

Funcionalidades típicas de edição tais como salvar, abrir um documento existente e criar novo documento também estão presentes no Peirce Online. O botão *New*, retorna os valores de  $\lambda_1$  e  $\lambda_2$  para os valores padrões definidos nesta dissertação, respectivamente, 0,5 e 5, e retorna os campos *Theory set*, *Conditions Set*, *Facts set* e *Hypotheses set* para {

}, para a realização de um novo raciocínio abduutivo. O botão *Open* é utilizado para carregar um arquivo texto (extensão .txt) contendo informações referentes a um modelo anteriormente salvo. A estrutura do arquivo texto (Figura 5.6) usa a linguagem Peirce Abdutiva, que será discutida na Seção 5.4:

```
Minimum Explanatory Power ( $\lambda_1$ ): 0,5
Maximum Complexity ( $\lambda_2$ ): 5

Theory set:
{
c  $\rightarrow$  m
}

Conditions set:
{

}

Facts set:
{
m
}

Hypotheses set:
{

}
```

**Figura 5.6: Exemplo de arquivo fonte escrito na linguagem Peirce Abdutiva. Este arquivo pode ser salvo e lido pelo sistema Peirce Online. Este arquivo também é transmitido entre o Cliente e o Servidor.**

O botão *Save* salva um modelo em um arquivo texto, no computador cliente, i.e., os dados contidos em  $\lambda_1$  e  $\lambda_2$  e nos campos *Theory set*, *Conditions set*, *Facts set* e *Hypotheses set*.

Proposições no Peirce Online podem ser escritas utilizando diferentes conjuntos de conectivos lógicos, alguns deles fáceis de serem digitados em teclados convencionais. A Tabela 5.1 descreve os diferentes símbolos que podem ser utilizados para representar os conectivos lógicos da LP.

Tabela 5.1: Símbolos representando conectivos lógicos que podem ser utilizados no Sistema Peirce Online.

<i>Conectivo lógico</i>	<i>Símbolos</i>
Negação	$\neg$ , !, $\sim$ , not, NOT
Conjunção	$\wedge$ , $\Lambda$ , and, AND, &
Disjunção	$\vee$ , $V$ , or, OR,
Implicação	$\Rightarrow$ , $\rightarrow$ , $\rightarrow$

## 5.2 – Arquitetura do Sistema Peirce Online

A Figura 5.7 descreve a arquitetura do Sistema Peirce Online.

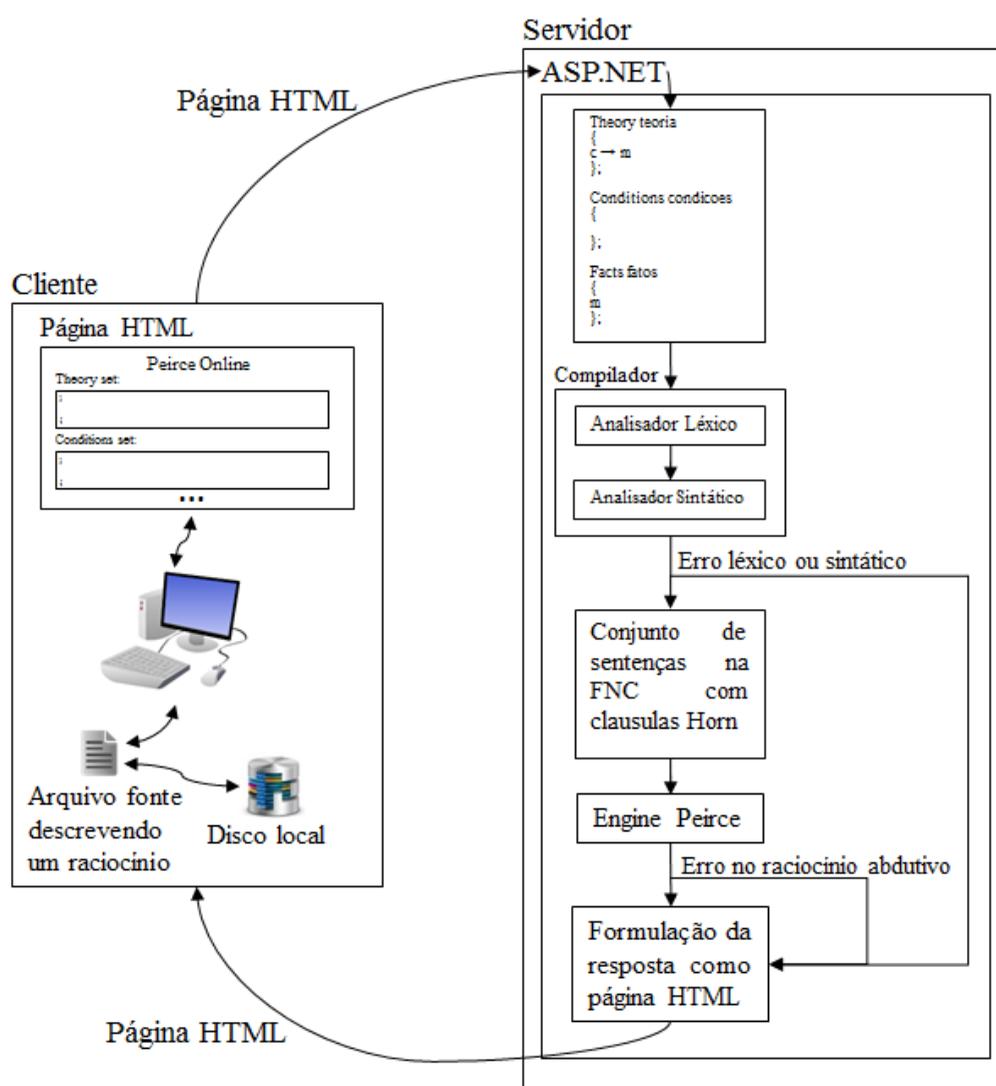


Figura 5.7: Arquitetura do Sistema Peirce Online.

O Sistema Peirce Online emprega uma arquitetura cliente-servidor. Na parte cliente páginas HTML permitem a interação com o sistema Peirce Online. O computador cliente pode utilizar arquivos fonte, que descrevem um raciocínio abduutivo, presentes em seu sistema de armazenamento local para que sejam abertos pelo sistema.

A parte cliente comunica-se com a parte servidor através do protocolo HTTP. A partir da página do Sistema onde se realizam raciocínios abdutivos o computador cliente se comunica com o computador servidor enviando a ele uma página HTML preenchida com  $\lambda_1$ ,  $\lambda_2$  e os campos *Theory set*, *Conditions set* e *Facts set*.

No servidor uma aplicação ASP.NET recebe a página HTML e prepara seu conteúdo para que as proposições na Forma HF sejam encaminhadas ao compilador. O compilador é composto de um analisador léxico e um analisador sintático. Ao compilar as proposições na Forma HF, vindas do cliente, o compilador gera proposições na FNC com cláusulas Horn e as transmite para a Engine Peirce que implementa o algoritmo Peirce (Capítulo 4). Após a realização do raciocínio abduutivo, pela Engine Peirce, uma página HTML é formulada como resposta e enviada ao cliente.

Caso ocorra algum erro na compilação ou na realização do raciocínio abduutivo, este erro é enviado ao cliente também como uma página HTML.

### **5.3 – Gramática da Linguagem Pierce Abduativa**

Chamamos de linguagem Peirce Abduativa a linguagem que é utilizada para descrever um modelo, i.e., os detalhes textuais que consolidam um raciocínio abduutivo no Sistema Peirce Online. A Figura 5.8 é um exemplo de um texto escrito na linguagem Peirce Abduativa.

```

Theory teoria
{
c → m
};

Conditions condições
{

};

Facts fatos
{
m
};

```

**Figura 5.8: Texto escrito na linguagem Peirce Abdutiva.**

A Figura 5.9 apresenta a gramática utilizada na produção de textos na linguagem Peirce Abdutiva. Esta linguagem está descrita com o formalismo EBNF (Capítulo 2, Definição 2.1). Neste exemplo no entanto aumentamos a fonte dos símbolos pertencentes à EBNF, como, por exemplo, as chaves, para que não sejam confundidos com os símbolos pertencentes à linguagem Peirce Abdutiva.

```

<programa> ::= <proposiçõesDeclarativas>

<proposiçõesDeclarativas> ::= <dclTeoria>
<dclCondição>
<dclFatos>

<dclTeoria> ::= Theory <identificador> {
<sequênciaProposiçõesHF> } ;

<sequênciaProposiçõesHF> ::= <proposiçãoHF> |
<proposiçãoHF> { , <sequênciaProposiçõesHF> }

<proposiçãoHF> ::= <proposiçãoConjuntiva> |
<proposiçãoDisjuntiva>

<proposiçãoConjuntiva> ::= <sequênciaProposiçõesComConjunção>
|
<sequênciaÁtomosComConjunção> →
<sequênciaProposiçõesComConjunção>

<sequênciaProposiçõesComConjunção> ::= <proposição> |
<proposição> { Λ <sequênciaProposiçõesComConjunção> }

<sequênciaÁtomosComConjunção> ::= <átomo> |
<átomo> { Λ <sequênciaÁtomosComConjunção> }

<proposiçãoDisjuntiva> ::=
<sequênciaÁtomosNegadosComDisjunção> |

```

```

<sequênciaProposiçõesComDisjunção> →
<sequênciaÁtomosNegadosComDisjunção>

<sequênciaÁtomosNegadosComDisjunção> ::= ¬<átomo> |
¬<átomo> { V <sequênciaÁtomosNegadosComDisjunção> }

<sequênciaProposiçõesComDisjunção> ::= <proposição> |
<proposição> { V <sequênciaProposiçõesComDisjunção> }

<proposição> ::= <átomo> | ¬<átomo>

<átomo> ::= <identificador>

<identificador> ::= <letra> | { <letra> | <dígito> |
<subtraço> }

<letra> ::= a | b | c | d | e | f | g | h | i | j | k | l | m
| n | o | p | q | r | s | t | u | v | w | x | y | z | A | B |
C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R
| S | T | U | V | W | X | Y | Z

<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<subtraço> ::= _

<dclCondições> ::= Conditions <identificador> { [
<sequênciaProposiçõesHF> ] } ;

<dclFacts> ::= Facts <identificador> { <sequênciaÁtomosFatos>
} ;

<sequênciaÁtomosFatos> ::= <átomo> |
<átomo> { , <sequênciaÁtomosFatos> }

```

**Figura 5.9: Gramática para produção de textos para o sistema Peirce Online.**

## 5.4 – O Compilador da Linguagem Peirce Abdutiva

O compilador da linguagem Peirce Abdutiva consiste em um analisador léxico e um analisador sintático. O Analisador léxico é um autômato finito determinístico (Hopcroft, Motwani & Ullman, 2006) que opera lendo quatro símbolos à frente. O analisador sintático foi desenvolvido utilizando a técnica de análise sintática descendente recursiva (Aho *et al.*, 2006). O Apêndice IV contém o pseudocódigo do compilador da linguagem Peirce Abdutiva.

# Capítulo 6

## Discussão e Conclusão

*Este capítulo apresenta discussões e conclusões desta dissertação. A Seção 6.1 discute a estrutura TCHF e as vantagens indicadas na utilização da mesma. A Seção 6.2 discute o Critério para a Seleção de Boas Hipóteses bem como as vantagens e desvantagens de sua utilização. A Seção 6.3 discute o algoritmo Peirce, seus pontos negativos e positivos em relação a realização de raciocínios abduativos e aos formalismos utilizados na concepção do mesmo. A Seção 6.4 discute a linguagem Peirce Abduativa e as diferenças entre a Forma HF e outras formas de restringir a escrita de proposições lógicas. A Seção 6.5 discute o sistema Peirce Online, os proveitos que podem ser obtidos de sua utilização e seus aspectos tecnológicos. A Seção 6.6 discorre sobre trabalhos futuros envolvendo outros critérios de seleção de boas hipóteses, outros mecanismos para a formulação de hipóteses candidatas e outras lógicas para a formalização do raciocínio abduativo a partir da estrutura TCHF.*

Este trabalho aborda o raciocínio abduutivo tendo a motivação daqueles que querem contribuir para o campo da Inteligência Artificial. Resultados de diferentes campos como a Psicologia Cognitiva, a Filosofia, os Fatores Humanos, entre outros, foram utilizados na compreensão dos fenômenos subjacentes ao raciocínio abduutivo e à proposta desta dissertação.

Por meio de trabalhos filosóficos e empíricos, de autores tradicionais e contemporâneos, o raciocínio abduutivo mostrou-se presente e importante para diversos campos de conhecimento e áreas de aplicação.

Neste trabalho foram pesquisadas estruturas para a formalização de raciocínios abduutivos, como a estrutura THF, e algoritmos já existentes que contemplavam a realização de raciocínios abduutivos, suas implementações computacionais e suas semelhanças com a realização de raciocínios abduutivos realizados por pessoas.

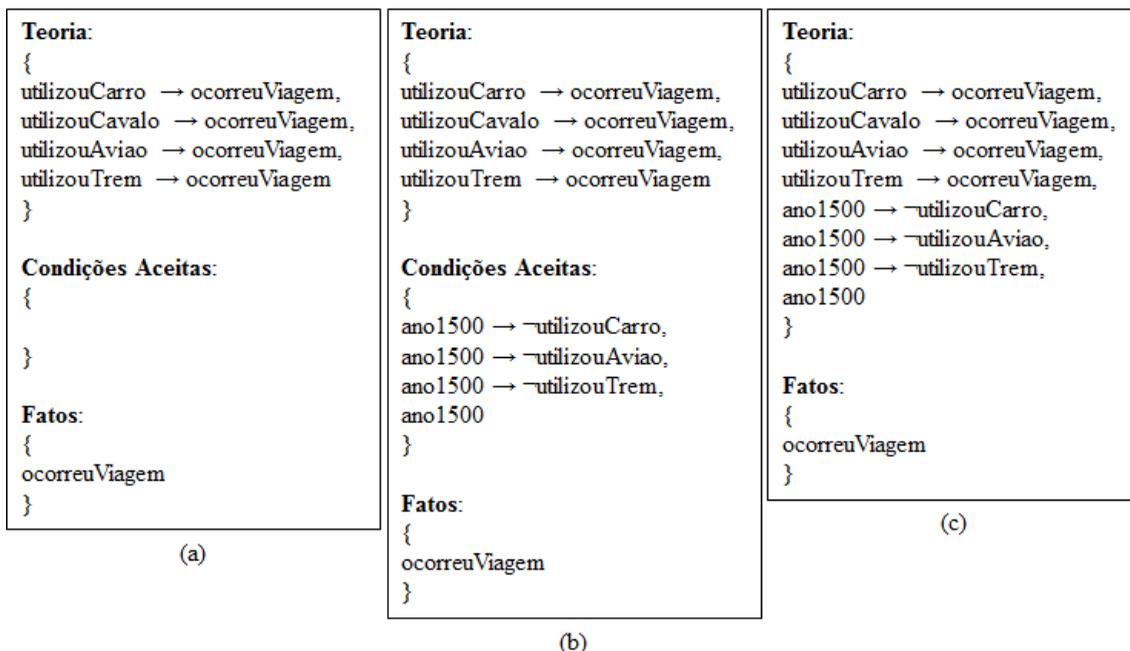
Com a pesquisa sobre raciocínio abduutivo em diferentes mídias e por meio da experimentação com estruturas e algoritmos existentes, notou-se a carência e a dificuldade na abordagem de importantes e recorrentes aspectos em raciocínios abduutivos: (1) a inclusão de condições que permeiam a realização de raciocínios abduutivos e, (2) a difícil tarefa de se estabelecer critérios para a seleção de boas hipóteses.

## **6.1 – Estrutura TCHF**

A utilização da estrutura TCHF sugeriu que esta abordagem apresenta aspectos positivos com relação à realização e à modelagem de raciocínios abduutivos. Ela oferece a possibilidade de utilizar conjuntos com um ou mais fatos, e isto se mostrou um facilitador para modelar diferentes cenários estudados e, em parte, relatados no Capítulo 1.

Esta estrutura possibilita também a especificação de conhecimentos que não fazem parte de uma teoria geral, evitando uma “contaminação” da teoria geral com especificidades contextuais, circunstanciais, intencionais etc. Por exemplo, a Figura 6.1-a apresenta uma teoria geral sobre viagens e meios de transporte. A Figura 6.1-b apresenta, no conjunto Condições Aceitas, condições que circunstanciam a teoria geral aos meios de transporte disponíveis no ano de 1500. Sem a presença do Conjunto Condições Aceitas na estrutura

de raciocínio a única maneira de modelar esta especificidade circunstancial seria adicionar proposições à teoria geral como apresenta a Figura 6.1-c.



**Figura 6.1: Diferentes maneiras de modelar uma condição circunstancial. (a) Apresenta uma teoria geral sobre meios de transportes para viajar; (b) circunstancia a teoria para o ano de 1500 fazendo uso do conjunto Condições Aceitas e; (c) “contamina” o conjunto Teoria com elementos circunstanciais relativos ao ano de 1500.**

A abordagem apresentada pela Figura 6.1-c provoca perda de generalidade do conjunto Teoria, fazendo com que a teoria fique presa às circunstâncias existentes do ano de 1500, por causa das novas proposições à ela adicionada, o mesmo problema ocorre em situações nas quais é necessário modelar os contextos, intenções etc. A abordagem ilustrada na Figura 6.1-c, sem o uso do conjunto de Condições Aceitas, exige, para uma mudança de contexto, a reformulação e reescrita da teoria.

O emprego do conjunto de Condições Aceitas também mostrou-se útil no estabelecimento de correlações lógicas entre diferentes teorias modeladas no conjunto Teoria da estrutura TCHF. O Exemplo 5.2, do Capítulo 5, apresenta o emprego do conjunto de Condições Aceitas para estabelecer correlações lógicas entre teorias.

## 6.2 – Critério para a Seleção de Boas Hipóteses

A escolha de critérios para a seleção de boas hipóteses não é uma tarefa trivial. Como discutido nesta dissertação, diversos autores propõem diferentes critérios para a seleção de boas hipóteses, não existindo um consenso.

A utilização dos critérios de força explanatória e complexidade na formulação de um critério único, que consiste na escolha de hipóteses com mínima complexidade entre as hipóteses candidatas com máxima força explanatória, é uma decisão de design de algoritmos fundamentada na opção por um critério simples que funciona relativamente bem na prática. A alta taxa de coincidência entre os raciocínios abduativos realizado por pessoas e pelo algoritmo Peirce, evidenciados pela pesquisa experimental realizada (Capítulo 4), sugere que esta escolha de design não foi ruim. Devemos observar, contudo, que o critério escolhido pode não funcionar bem em muitas situações e que alternativas para ele existem e podem ser propostas.

### 6.3 – Algoritmo Peirce

A Lógica Proposicional foi escolhida como formalismo para a representação, manipulação de conhecimento e cálculo de hipóteses candidatas no algoritmo Peirce. Esta escolha foi motivada pela simplicidade desta lógica, que reflete na facilidade para modelar raciocínios, a não necessidade de mensuração e incorporação de parâmetros aos modelos e a descomplicação do mecanismo de resolução.

Como tudo, esta escolha tem suas desvantagens sendo, a principal delas, a limitação do poder expressivo da Lógica Proposicional por exemplo quando comparada com a Lógica de Primeira Ordem (ou Lógica de Predicados).

Outro aspecto, não necessariamente uma desvantagem, tem relação com a formulação de hipóteses. Em um conjunto de proposições  $a \rightarrow b$ ,  $b \rightarrow c$ , a ocorrência do fato  $c$  faz com que o algoritmo calcule a hipótese  $a$ , embora  $b$  também seja uma hipótese aceitável. O Exemplo 6.1 ilustra esta funcionalidade.

**Exemplo 6.1.** Propomos aqui um modelo de uma teoria que contém explicações referentes à causa da morte de um esquiador, conforme as proposições:

- *colisao\_pedra*: Houve colisão com pedra.
- *fratura\_cranio*: Ocorreu fratura no crânio.
- *aumento\_pressao\_encefalica*: Ocorreu aumento da pressão encefálica.
- *morte\_esquiador*: ocorreu a morte do esquiador.

Os conjuntos Teoria, Condições Aceitas e Fatos, para este exemplo, são definidos como na Figura 6.2-a-b-c, e após a execução de Peirce as hipóteses geradas são as apresentadas na Figura 6.2-d:

**Teoria:**  
{  
colisao\_pedra → fratura\_cranio,  
fratura\_cranio → aumento\_pressao\_encefalica,  
aumento\_pressao\_encefalica → morte\_esquiador  
}

(a)

**Condições Aceitas:**  
{  
  
}

(b)

**Fatos:**  
{  
morte\_esquiador  
}

(c)

**Hipóteses:**  
{  
colisao\_pedra  
}

(d)

**Figura 6.2: Exemplo de um raciocínio abduutivo para explicar as causas da morte de um esquiador.** Neste conjunto Teoria temos como explicações para o fato *morte\_esquiador* as hipóteses *colisao\_pedra*, *aumento\_pressao\_encefalica* e *fratura\_cranio*. Entretanto, como pode-se observar na Figura 6.2-d, o algoritmo formula apenas a hipótese *colisao\_pedra*.

Por razões relacionadas ao mecanismo de resolução, o algoritmo Peirce formula sempre as hipóteses mais “distantes” dos fatos, neste exemplo *colisao\_pedra*. No entanto, neste exemplo, diferentes visões e interesses poderiam influenciar a seleção de hipóteses menos distantes dos fatos. Em um cenário médico-acadêmico é provável que *fratura\_cranio* e/ou *aumento\_pressao\_encefalica*, sejam as hipóteses escolhidas e não a hipótese *colisao\_pedra*.

□

Cabe observar que a funcionalidade descrita anteriormente não é exatamente um problema do algoritmo Peirce em si. O conjunto de Condições Aceitas pode ser utilizado

para descrever proposições que discriminam hipóteses mais aceitáveis e menos aceitáveis em uma situação ou contexto específico.

Outro aspecto que deve ser destacado sobre o algoritmo Peirce é seu tempo de execução polinomial.

#### **6.4 – Linguagem Peirce Abdutiva**

A linguagem Peirce Abdutiva opera sobre proposições pertencentes à linguagem da LP na Forma HF. Embora a Forma HF restrinja a escrita de sentenças da LP, ela confere mais liberdade de escrita do que muitas soluções existentes. Por exemplo, caso fosse semelhante a Prolog, a Forma HF incluiria apenas construções sintáticas do tipo  $a_1 \wedge a_2 \wedge a_3 \dots \wedge a_n \rightarrow c$ , onde  $a_i$  ( $1 \leq i \leq n$ ) são literais e  $c$  é um átomo. Este tipo de construção sintática, por exemplo, não permite que ao lado direito de uma implicação contenha um átomo negado ou então apresente vários literais conectados pelos conectivos  $\wedge$  ou  $\vee$ .

A linguagem Peirce Abdutiva oferta também outra comodidade. Ela possibilita a utilização de proposições cíclicas, como em  $a \rightarrow b, b \rightarrow a$ . Sistemas como Prolog normalmente não permitem construções cíclicas.

#### **6.5 – O Sistema Peirce Online**

O Sistema Peirce Online permite a criação de modelos de fenômenos, máquinas, situações do cotidiano, conhecimento especialista, etc. Estes modelos podem ser estruturados por diversas teorias e condições.

De forma geral, o Sistema Peirce Online pode ser útil para auxiliar pessoas a realizarem raciocínios, mas também pode ser empregado com objetivos específicos. A seguir nós discutimos como o Sistema pode ser empregado em um contexto educacional Construcionista (Papert, 1980) e também para validação de uma teoria científica.

No campo educacional o processo de desenvolvimento de modelos pode ser utilizado como ferramenta para que estudantes aprendam sobre um domínio por meio da atividade de modelar, executar um modelo, refletir sobre o resultado da execução e, possivelmente, reformular o modelo para que ele reflita paulatinamente o domínio de interesse. Por exemplo, considere que o Modelo 1 do circuito elétrico apresentado na

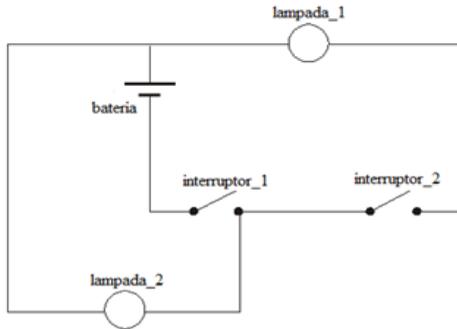
Figura 6.3-a é uma representação de um circuito real. A execução deste modelo na ocorrência do conjunto de fatos  $F = \{lampada\_1\_acesa, lampada\_2\_apagada\}$  produz como resposta um conjunto de hipóteses vazio, ou seja, o Sistema não consegue explicar o conjunto de fatos F. Entretanto, é plenamente possível ocorrer tal conjunto de fatos no circuito real que motiva o Modelo 1. Isto pode levar um estudante a reformular o Modelo 1 para incluir novos fenômenos relativos a circuitos elétricos. A Figura 6.3-b apresenta o Modelo 2 como uma possível reformulação do Modelo 1. O Modelo 1 trata os fios que conectam os dispositivos como uma única peça. Em contrapartida, o Modelo 2 introduz a ideia de segmentos de diversos fios conectando os dispositivos do circuito. A execução do Modelo 2 na ocorrência do conjunto dos mesmos fatos F, descritos acima, agora produz o conjunto de hipóteses:

$$H = \{ \text{bateria\_com\_carga} \wedge \text{fio\_1\_rompido} \wedge \text{fio\_3\_conectado} \wedge \text{fio\_4\_conectado} \\ \wedge \text{fio\_5\_conectado} \wedge \text{fio\_6\_conectado} \wedge \text{interruptor\_1\_ligado} \wedge \\ \text{interruptor\_2\_ligado},$$

$$\text{bateria\_com\_carga} \wedge \text{fio\_2\_rompido} \wedge \text{fio\_3\_conectado} \wedge \text{fio\_4\_conectado} \wedge \\ \text{fio\_5\_conectado} \wedge \text{fio\_6\_conectado} \wedge \text{interruptor\_1\_ligado} \wedge \text{interruptor\_2\_ligado} \\ \}.$$

O Modelo 2, mais aprimorado do que o Modelo 1, contém uma explicação para o conjunto de fatos F. A reformulação do Modelo 1 para o Modelo 2 pelo estudante dá a ele uma oportunidade para aprender sobre circuitos.

Modelo 1:



```

Theory set:
{
// Teoria 1: Teoria sobre explicações que poderiam levar
//as lâmpadas do circuito estarem apagadas.
bateria_sem_carga → lampada_1_apagada ∧
lampada_2_apagada,
fios_rompidos → lampada_1_apagada ∧ lampada_2_apagada,
interruptor_1_desligado → lampada_1_apagada ∧
lampada_2_apagada,
interruptor_2_desligado → lampada_1_apagada,

// Teoria 2: Teoria sobre explicações que poderiam levar
//as lâmpadas do circuito estarem acesas.
bateria_com_carga ∧ fios_conectados ∧ interruptor_1_ligado
→ lampada_2_acesa,
bateria_com_carga ∧ fios_conectados ∧ interruptor_1_ligado
∧ interruptor_2_ligado → lampada_1_acesa ∧
lampada_2_acesa
}

Conditions set:
{
// Proposições descrevendo correlações lógicas entre
// proposições da Teoria 1 e da Teoria 2.
bateria_com_carga → ¬bateria_sem_carga,
interruptor_1_desligado → ¬interruptor_1_ligado,
interruptor_2_desligado → ¬interruptor_2_ligado,
fios_rompidos → ¬fios_conectados
}

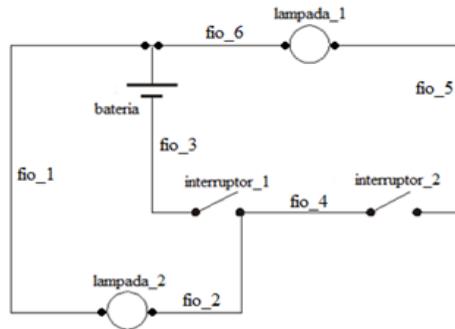
Facts set:
{
lampada_1_acesa, lampada_2_apagada
}

Hypotheses set:
{
}

```

(a)

Modelo 2:



```

Theory set:
{
// Teoria 1: Teoria sobre explicações que poderiam levar
//as lâmpadas do circuito estarem apagadas.
bateria_sem_carga → lampada_1_apagada,
bateria_sem_carga → lampada_2_apagada,
fio_1_rompido → lampada_2_apagada, fio_2_rompido →
lampada_2_apagada, fio_3_rompido → lampada_2_apagada,
fio_3_rompido → lampada_1_apagada, fio_4_rompido →
lampada_1_apagada, fio_5_rompido → lampada_1_apagada,
fio_6_rompido → lampada_1_apagada,
interruptor_1_desligado → lampada_1_apagada,
interruptor_1_desligado → lampada_2_apagada,
interruptor_2_desligado → lampada_1_apagada,

// Teoria 2: Teoria sobre explicações que poderiam levar
//as lâmpadas do circuito estarem acesas.
bateria_com_carga ∧ fio_2_conectado ∧ fio_1_conectado ∧
fio_3_conectado ∧ interruptor_1_ligado → lampada_2_acesa,
bateria_com_carga ∧ fio_3_conectado ∧ fio_4_conectado ∧
fio_5_conectado ∧ fio_6_conectado ∧ interruptor_1_ligado ∧
interruptor_2_ligado → lampada_1_acesa
}

Conditions set:
{
// Proposições descrevendo correlações lógicas entre
// proposições da Teoria 1 e da Teoria 2.
bateria_com_carga → ¬bateria_sem_carga,
interruptor_1_desligado → ¬interruptor_1_ligado,
interruptor_2_desligado → ¬interruptor_2_ligado,
fio_1_rompido → ¬fio_1_conectado,
fio_2_rompido → ¬fio_2_conectado,
fio_3_rompido → ¬fio_3_conectado,
fio_4_rompido → ¬fio_4_conectado,
fio_5_rompido → ¬fio_5_conectado,
fio_6_rompido → ¬fio_6_conectado
}

Facts set:
{
lampada_1_acesa, lampada_2_apagada
}

Hypotheses set:
{
bateria_com_carga ∧ fio_1_rompido ∧ fio_3_conectado ∧
fio_4_conectado ∧ fio_5_conectado ∧ fio_6_conectado ∧
interruptor_1_ligado ∧ interruptor_2_ligado,
bateria_com_carga ∧ fio_2_rompido ∧ fio_3_conectado ∧
fio_4_conectado ∧ fio_5_conectado ∧ fio_6_conectado ∧
interruptor_1_ligado ∧ interruptor_2_ligado
}
}

```

(b)

Figura 6.3: Exemplo da evolução de um modelo.

O Sistema Peirce Online também poderia ser utilizado para verificar a validade de uma teoria na presença de novas observações. Por exemplo, a Teoria do Flogisto para combustão, formulada em 1667 por Johann Becker (Thagard, 1989), está parcialmente representada no Modelo 1, apresentado na Figura 6.4-a. Segundo a Teoria do Flogisto, corpos contêm Flogisto que, na combustão, são despreendidos. A execução do Modelo 1 para o fato *corpoQueimado* encontra como explicação a hipótese *corpoComFlogisto  $\wedge$  flogistoDespreendido*.

Mais tarde, Lavoisier observou que, na combustão, os corpos queimados apresentam um aumento de massa. O Modelo 2, apresentado na Figura 6.4-b, introduz no conjunto de fatos do Modelo 1 a observação de Lavoisier: no conjunto Teoria foi incluída uma proposição sobre a diminuição ou não da massa de um corpo em decorrência do desprendimento de Flogisto e, no conjunto de Condições foi incluído uma proposição sobre a inconsistência entre o aumento e a diminuição ou conservação da massa do corpo antes e depois da combustão. A execução do Modelo 2, que captura a observação de Lavoisier, invalida a Teoria do Flogisto na medida que não é possível encontrar explicações para o conjunto de fatos (conjunto de hipóteses vazio). Uma nova Teoria precisa ser formulada e, historicamente, foi o que fez Lavoisier.

<pre> <b>Modelo 1</b> Theory set: { // Teoria do Flogisto corpoComFlogisto <math>\wedge</math> flogistoDesprendido <math>\rightarrow</math> corpoQueimado }  Conditions set: {  }  Facts set: { corpoQueimado }  Hypotheses set: { corpoComFlogisto <math>\wedge</math> flogistoDesprendido } </pre> <p style="text-align: center;">(a)</p>	<pre> <b>Modelo 2</b> Theory set: { // Teoria do Flogisto corpoComFlogisto <math>\wedge</math> flogistoDesprendido <math>\rightarrow</math> corpoQueimado,  // Proposição acrescentada em decorrência da observação de Lavoisier. flogistoDesprendido <math>\rightarrow</math> massaCorpoQueimadoMenorOuIgualQueMassaCorpo }  Conditions set: { // Proposições correlacionando a inconsistência entre o aumento e a // diminuição ou conservação da massa do corpo antes e depois da // combustão massaCorpoQueimadoMenorOuIgualQueMassaCorpo <math>\rightarrow</math> <math>\neg</math>massaCorpoQueimadoMaiorQueMassaCorpo }  Facts set: { corpoQueimado,  // Observação de Lavoisier: massa de corpo queimado é maior do que // a do corpo antes da combustão. massaCorpoQueimadoMaiorQueMassaCorpo }  Hypotheses set: {  } </pre> <p style="text-align: center;">(b)</p>
---	--

**Figura 6.4: Exemplo de verificação de validade de uma teoria.**

Com relação ao uso de técnicas de compilação, o emprego de Análise Sintática Descendente Recursiva (Aho, Sethi e Ullman, 2006) facilitou a prototipação do compilador em um momento em que não se tinha uma noção clara de como ficaria a gramática da linguagem Peirce Abdutiva, evitando sobrecarga de trabalho em compor e recompor tabelas de símbolos para, por exemplo, Análise Sintática LR(k).

## 6.6 – Trabalhos Futuros

A investigação de critérios para a seleção de boas hipóteses é um campo aberto. Pode-se conjecturar que existam critérios gerais capazes de dar conta da seleção entre hipóteses candidatas, independentemente de domínios. Esta é uma possível linha para formulação de trabalhos futuros.

Em contrapartida, também pode-se conjecturar que a seleção de hipóteses é dependente de domínios e contextos, configurando outra linha de investigação.

A linguagem Peirce Abdutiva permite a descrição de modelos constituídos por um conjunto Teoria, um conjunto de Condições Aceitas e um conjunto de Fatos. Isto tem se

mostrado útil para desenvolver modelos. No entanto, para o desenvolvimento de modelos maiores constituídos, eventualmente por várias teorias, condições e fatos é necessária uma extensão da linguagem Peirce Abdutiva. É um futuro trabalho o desenvolvimento de uma linguagem que permita a declaração nominal de várias teorias, condições, fatos e, também, de raciocínios abdutivos. O resultado de um raciocínio, da forma como definimos, é um conjunto. A inclusão na linguagem do recurso de avaliação de expressões envolvendo operações de união, intersecção e subtração de raciocínios abrirá espaço para “programação” de raciocínios mais complexos como, por exemplo, raciocínios que se desenvolvem a partir da oposição dialética entre dois raciocínios mais elementares.

A adaptação do algoritmo Peirce para Lógica de Primeira Ordem é também um interessante desafio para um trabalho futuro.

Finalmente, gostaríamos de observar que o estudo do raciocínio abductivo vem recebendo bastante atenção nos últimos tempos, devido à importância deste tipo de raciocínio para diversas áreas, dentro e fora do âmbito da Computação. Este trabalho procurou capturar aspectos fundamentais do raciocínio abductivo bem como as formalizações necessárias ao estudo do assunto sob a perspectiva computacional. Observando a necessidade de um complemento às estruturas existentes, devido à presença de características inerentes e importantes do raciocínio abductivo que foram pouco exploradas nos trabalhos analisados, este trabalho propôs uma ampliação à estrutura comumente abordada na Lógica Proposicional. Apesar do foco deste trabalho estar na Lógica as abordagens aqui propostas podem ser aplicadas à quaisquer formalizações de raciocínios abductivos, seja para Programação Lógica, Redes Bayesianas, Modelos Conexionistas, Problemas de Buscas, etc., esta ampliação sugeriu-se positiva à medida que foi contrastada outras com estruturas e exemplos envolvendo raciocínios abductivos, e ao ser contrastada com raciocínios abductivos realizados por seres humanos. Um ponto importante para se destacar é a implementação do algoritmo Peirce para realização de raciocínios abductivos em um ambiente Web, quebrando as limitações de distribuição de softwares desktop referentes tanto à Sistema Operacional, quanto à distribuição do Sistema.

# Apêndice I

## Mais cenários exemplo do uso do raciocínio abduativo

### Cenário I.1 – Investigação policial (adaptado de um caso real)

Um policial investigador chega à cena de um crime onde um casal foi morto com golpes na cabeça, enquanto dormia à noite. Uma jarra com um pouco de água está sobre o criado mudo ao lado da mulher, mas não há copo, o que faz o investigador supor que a jarra não foi levada para o quarto pelo casal. A mulher tem, colocada na sua boca, uma toalha. As armas que foram usadas não estão presentes na cena do crime. O investigador examina os demais recintos da residência. A casa não apresenta sinais de arrombamento e está bem arrumada. Apenas uma pasta contendo algum dinheiro foi rasgada, provavelmente com o auxílio de uma faca, e encontra-se jogada ao chão. O investigador descobre que a jarra pertence à casa pois faz parte de um conjunto em que falta somente ela. A toalha também é da casa e foi retirada de um conjunto de toalhas semelhantes. Todas as outras toalhas e peças do conjunto a que pertence a jarra estão bem arrumadas, o que faz o investigador supor que quem tirou a toalha e a jarra sabia onde elas estavam. O casal tinha um filho, ainda menino, e uma filha de 19 anos que disseram ter chegado em casa e encontrado os pais mortos. A filha disse que à noite estava com o namorado em um motel e apresentou uma nota fiscal do estabelecimento, para surpresa do investigador, afinal não é comum usuários solicitarem notas fiscais de motel. O filho à noite estava em um *shopping center*, o que foi confirmado, e retornou para a casa no carro dirigido pela irmã. Baseado nestas evidências o investigador formulou a hipótese da filha ter tido participação criminosa na morte dos pais. O fato do casal ter sido assassinado simultaneamente com fortes golpes na cabeça enquanto dormia fez o investigador raciocinar que a filha não poderia, pelo seu porte físico, ter participado sozinha do crime que levou à morte os seus pais. ...

Este cenário sugere que o trabalho do policial investigador é o de tentar desenvolver hipóteses sobre como se deu um crime e quem são os criminosos, a partir das evidências (fatos) da cena do crime e do conhecimento teórico que ele possui sobre relações sociais, cultura, hábitos das famílias, funcionamento de uma residência familiar e *modus operandi* de homicídios, entre outros. Investigações como esta são complexas e na prática correspondem a sucessivos raciocínios abduativos, à medida em que o investigador vai incorporando mais e mais evidências. O raciocínio abduativo condicional, isto é, com aceitação ou não da verdade de certas condições é utilizado com muita frequência e intensidade. Por exemplo, neste cenário o investigador raciocina para

compreender o crime admitindo como condição que a jarra e a toalha foram levados para a cena do crime pelos criminosos e que quem pegou a toalha e a jarra sabia onde eles se encontravam.

Este cenário é uma adaptação de um caso real. Aliás, constitui também um exemplo de raciocínio abduativo tentar formular uma hipótese sobre qual caso real de investigação corresponde a este.

### **Cenário I.2 – Interpretação de texto**

Raciocínios abduativos desempenham um importante papel na interpretação de textos. Palavra após palavra, sentença após sentença, frase após frase, parágrafo após parágrafo e capítulo após capítulo, novos fatos são revelados em um texto. À medida em que o leitor vai lendo, raciocínios abduativos são sucessivamente desenvolvidos e aprimorados por ele. O presente cenário é o primeiro parágrafo da obra literária “O Seminarista” de Rubem Fonseca (2009) e corresponde a uma narrativa do protagonista, o Seminarista.

Sou conhecido como o Especialista, contratado para serviços específicos.

Entre outros raciocínios, o leitor poderia raciocinar para formular hipóteses sobre quais são a especialidade e o serviço feito pelo protagonista: especialista em ritos sagrados, exorcismo, um matador, outras inúmeras hipóteses.

O Despachante diz quem é o freguês, me dá as coordenadas e eu faço o serviço.

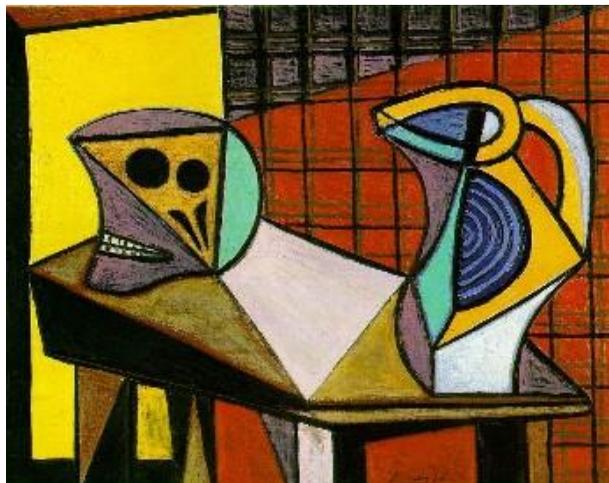
Nesta sentença novos fatos são apresentados: a especialidade tem a ver com serviços que são feitos com uma pessoa, o freguês. As expressões “freguês” como referência àquele com o qual o serviço deve ser feito, e “despachante” como apelido e papel do intermediário que define o serviço, enfraquece possíveis hipóteses de especialidades religiosas e fortalece hipóteses que têm relação com o crime. Matadores, traficantes etc. ganham força como possíveis hipóteses.

Antes de entrar no que interessa – Kirsten, Ziff, D.S., Sangue de Boi – eu vou contar como foram alguns dos meus serviços.

A referência aos serviços pelo nome de pessoas fortalece ainda mais hipóteses como a de ser o protagonista um matador.

### Cenário I.3 – Interpretação de pinturas e imagens

A Figura I.1 é este cenário.



**Figura I.1: Uma pintura abstrata.**

Na interpretação de figuras abstratas tenta-se estabelecer significado para elas. E nesta tentativa é onde fundamentalmente se encontra o raciocínio abduutivo. Tratam-se de raciocínios complexos por que, em geral, muitas hipóteses sobre o significado de uma pintura abstrata podem ser estabelecidas.

A maior graça deste tipo de pintura está neste difícil processo de raciocínio abduutivo, que situa-se entre o decifrar uma possível e jamais concretamente revelada mensagem do pintor atingindo um ponto no qual o apreciador, no limite das suas forças para raciocinar, se dá por satisfeito pela quase coerência do todo.

Especificamente, no caso da pintura da Figura I.1, os raciocínios abdutivos para estabelecimento de um significado a ela ficariam mais simples se novas informações fossem acrescentadas. Ela foi pintada em 1945 por Pablo Picasso e tem como título em Inglês *Skull and Pitcher*. O ano e o autor oferecem um contexto que restringe o estabelecimento de hipóteses, criando facilidades para o raciocínio. O título (fato) soma-se às várias formas (fatos) presentes na pintura, adicionando pistas que também contribuem para facilitar o raciocínio.

# Apêndice II

## Algoritmos para Abdução baseados em Lógica

Este apêndice discute algoritmos formalizados por meio de lógica para a realização de raciocínios abduativos. A Seção II.1 introduz o Tableau Semântico, um método com várias utilizações, em especial para Prova de Teoremas. A Seção II.2 mostra como o método dos Tableaux Semânticos pode ser usado para a realização de raciocínios abduativos. A Seção II.3 apresenta um algoritmo de construção e análise de tableaux semânticos para raciocínio abduativo. A Seção II.4 aborda a Programação Lógica Abduativa.

### II.1 – Tableau Semântico

Apresentado nos anos 50 por Evert Willem Beth, o Tableau Semântico é amplamente utilizado em lógica (Aliseda, 2006), permitindo em especial a Prova de Teoremas, entre outras utilizações.

A representação de proposições pertencentes à linguagem da LP através do Tableau Semântico é feita a partir de uma árvore utilizando as construções apresentadas na Tabela II.1

Tabela II.1: Representações de proposições na estrutura de árvore do Tableaux Semântico.

<i>Proposição</i>	<i>Em Lógica Proposicional</i>	<i>Representação no Tableaux Semântico</i>
<i>Átomo</i>	$a$	a
<i>Dupla Negação</i>	$\neg\neg c$	c
<i>Conjunção:</i>	$a \wedge c$	$\begin{array}{c} a \\   \\ b \end{array}$
<i>Disjunção:</i>	$a \vee b$	$\begin{array}{cc} & \wedge \\ / & \backslash \\ a & b \end{array}$
<i>Implicação:</i>	$a \rightarrow b$	$\begin{array}{cc} & \wedge \\ / & \backslash \\ \neg a & b \end{array}$

A utilização do Tableaux Semântico é também similar à Prova por Contradição. Ou seja, se queremos provar que  $\alpha \models \beta$ , dado que  $\alpha = (p \rightarrow q) \wedge p$  e  $\beta = q$ , podemos comprovar esta consequência lógica se obtivermos uma contradição a partir de  $(p \rightarrow q) \wedge p \wedge \neg q$ .

A Figura II.1 apresenta a representação em árvore das proposições  $\alpha$  e  $\neg\beta$  utilizando o Tableaux Semântico.

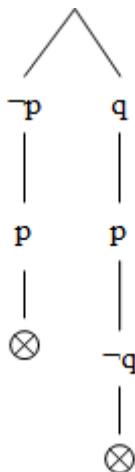


Figura II.1: Representação em árvore das proposições  $\alpha = (p \rightarrow q) \wedge p$  e  $\beta = q$  utilizando Tableaux Semântico.

No tableau da Figura II.1 pode-se observar o símbolo  $\otimes$ , este símbolo representa que o ramo encontra-se fechado contendo um átomo e sua negação, neste exemplo temos no ramo da esquerda  $\neg p$  e  $p$  e no ramo da direita temos  $q$  e  $\neg q$ . Cada ramo da árvore representa uma conjunção das proposições presentes no ramo. A árvore em sua totalidade representa uma disjunção dos ramos.

Ao analisar a árvore da Figura II.1 do ramo da esquerda ao ramo da direita obtém-se a seguinte sentença que pode ser interpretada como uma sentença na FND:

- $(\neg p \wedge p) \vee (q \wedge p \wedge \neg q)$ .

Esta sentença é uma contradição, pois é uma disjunção de proposições cuja interpretação é F, verificando assim que  $\alpha \models \beta$ . Nota-se também, com a relação ao tableau da Figura II.1, que o tableau está completamente fechado indicando que a sentença utilizada na montagem do tableau é uma contradição.

Após a montagem, um tableau semântico pode apresentar as seguintes propriedades:

- Ramo fechado: O ramo contém um átomo e sua negação.
- Ramo aberto: O ramo contém átomos, mas não apresenta a negação de nenhum átomo contido nele.
- Tableau fechado: A árvore apresenta todos os ramos fechado.
- Tableau aberto: A árvore apresenta pelo menos um ramo aberto.

A interpretação dos ramos e do tableau depende das definições utilizadas na resolução do problema à que se aplica o Tableaux Semântico. A próxima seção apresenta interpretações para as propriedades do tableau de acordo com o raciocínio abdutivo.

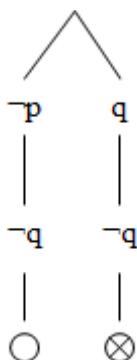
## II.2 – Tableaux Semântico para abdução

Esta seção apresentará como o Tableaux Semântico pode ser utilizado para a realização de raciocínios abduativos. A formalização e técnica para a utilização do Tableaux Semântico para a realização de raciocínios abduativos são baseadas no estudo de Aliseda (2006).

De acordo com Aliseda o Tableau Semântico permite com facilidade a obtenção de proposições atômicas, conjuntivas e disjuntivas, ao realizar raciocínios abduativos, devido a sua estrutura e as estratégias que podem ser empregadas para interpretá-la.

Utilizaremos aqui a estrutura de raciocínio THF, referentes à: um conjunto de proposições representando o conjunto teoria (T), um átomo representando o fato observado (F), e um conjunto de hipótese (H) que explicam o fato observado de acordo com a teoria. Para a realização de um raciocínio abduativo é preciso formular um conjunto H que juntamente com T explique F e que atenda as condições conforme as expressões apresentadas no Capítulo 2, Seção 2.2. O Exemplo II.1 apresenta como é a realização de um raciocínio abduativo através do Tableau Semântico.

**Exemplo II.1.** Considerando os conjuntos  $T = \{ p \rightarrow q \}$ ,  $F = \{ q \}$ , monta-se o tableau de  $p \rightarrow q \wedge \neg q$ , Figura II.2:



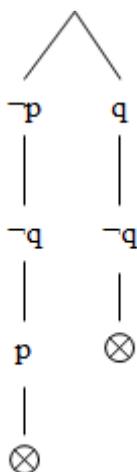
**Figura II.2:** Tableau semântico referente à proposição  $p \rightarrow q \wedge \neg q$ .

Desta vez nota-se o símbolo  $\bigcirc$  indicando que o ramo correspondente do tableau está aberto. Que para um raciocínio abduativo isso significa que a partir do conjunto T, o fato apresentado no conjunto F não é explicado, sendo assim é preciso buscar um conjunto de proposições para serem agregadas ao ramo, estas proposições compõem o conjunto de hipóteses H.

Para a geração destas proposições observa-se quais proposições compõem o ramo aberto do tableau e procura-se proposições para fecha-lo. Como definido na Seção II.1, para o fechamento de um ramo do tableau, este deve apresentar um átomo e sua negação. Para o ramo aberto deste exemplo temos as proposições  $\neg p$  e  $\neg q$ , um conjunto de proposições que fechariam esse ramo poderia ser composto de, por exemplo,  $p$ ,  $q$ ,  $p \wedge q$ , destas e de

outras proposições, são selecionadas não conflitantes com as condições da estrutura THF e compõe-se o conjunto H, de hipóteses abdutivas.

Escolhendo, por exemplo, a proposição  $p$  obtém-se o tableau da Figura II.3.



**Figura II.3: Tableau fechado com a inserção da nova proposição  $p$ , significando que um raciocínio abduativo foi realizado e esta nova proposição é uma hipótese abduativa.**

Com a proposição  $p$  fechando o tableau, estamos diante de  $T \cup \{ p \} \wedge \neg F$  ser uma contradição, ou seja,  $T \cup H \models F$  caracterizando que  $p$  é uma hipótese candidata ao conjunto de hipóteses H.

□

### II.3 – Algoritmo para raciocínios abdutivos de Aliseda baseado em Tableaux Semântico

No intuito de automatizar a realização de raciocínios abdutivos utilizando o Tableaux Semântico Aliseda (2006) apresenta um algoritmo para a realização de raciocínios abdutivos utilizando a estrutura do Tableaux Semântico.

Além das condições da estrutura THF já apresentadas, as formulações de hipóteses pelo algoritmo proposto também seguem as seguintes condições: vocabulário, forma e não redundância.

Através da condição de vocabulário as hipóteses geradas devem estar dentro dos elementos relacionados à teoria e ao fato observado, desta maneira se temos  $h$  como hipótese,  $h$  deve pertencer ao vocabulário de T e F, apresentando em sua proposição átomos ou átomos negados que pertençam a T e F, i.e.,  $h \in \text{Voc}(T \cup F)$ .

A condição de forma implica que uma hipótese  $h$  ou é um átomo ou átomo negado, ou uma conjunção de proposições sem átomos repetidos ou uma disjunção de proposições sem átomos repetidos.

A condição de não redundância implica que para uma hipótese  $h$  não ser redundante se ela é atômica ou nenhuma proposição de  $h$ , diferente de F, é uma hipótese que explica os fatos, i.e.,  $h$  deve ser mínima.

Por exemplo:

- Tem-se o fato  $q$ .
- Considerando o conjunto de possíveis hipóteses  $H$  que explicam  $q$ :  $H = \{ p, p \wedge r, q \}$ .

De acordo com a condição de não redundância, apenas a hipótese  $p$  seria escolhida dentre todas as apresentadas no conjunto  $H$ . Isto se dá por: (1) tanto a hipótese  $p$  quanto a hipótese  $p \wedge r$  explicam  $q$ , porém  $p$  é uma proposição de  $p \wedge r$  que explica  $q$  sendo assim  $p \wedge r$  é considerada redundante e descartada; (2) a hipótese  $q$  apesar de explicar  $q$ , é o próprio fato, o que também faz com que a hipótese  $q$  seja redundante e descartada.

A Figura II.4 apresenta uma interpretação do algoritmo para formular hipóteses abduativas a partir do Tableaux Semântico.

```

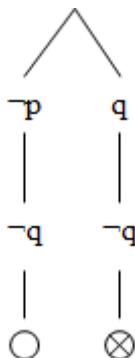
Algoritmo Abducao_Utilizando_Tableaux_Semantico(T, F):
Entrada:
  • Um conjunto de proposição pertencentes à linguagem da LP
    representando uma teoria T.
  • Um conjunto contendo um átomo representando o fato observado.
  • Precondições:  $T \neq F$ ,  $T \neq \neg F$ .
Saída:
  • Conjunto H de hipóteses que fecham o Tableau  $(T \cup \neg F)$ , e
    atendem as condições de vocabulário, forma e redundância, ou
    um conjunto vazio representando que não há hipóteses que
    expliquem o fato.
35 {
36   A := Desenvolver o Tableaux de  $T \cup \neg F$ ;
37   B := Computar o conjunto de ramos abertos de A tal que
      cada  $b \in B$  é um ramo aberto;
38   H := Proposições  $h$  tal que cada  $h \in H$  fecha todos os
      ramos abertos de B e  $h \neq f$ ;
39   se  $H = \{\}$  então
40     {
41       para  $i = 1$  até  $|B|$  faça
42         {
43            $H' :=$  Cada  $h' \in H'$  é um conjunto, tal que cada
            elemento é uma proposição que fecha um ramo
             $b_i$ , mas não fecham o tableau;  $h_1'$  é um
            conjunto de proposições que fecha  $b_1$ ,  $h_2'$  é
            um conjunto de proposições que fecha  $b_2$ ,  $h_3'$ 
            é um conjunto de proposições que fecha  $b_3$ ,
            etc.
44         }
45       H := Conjunções de uma proposição de cada conjunto de
         H'      tal que cada  $h \in H$  fecha o tableau;
46     }
47   retornar H.
48 }

```

**Figura II.4: Algoritmo para abduções utilizando Tableaux Semântico.**

Primeiramente o algoritmo constrói a árvore representando o Tableau semântico de  $T \cup \neg F$  e o armazena na estrutura A (linha 2). Em seguida forma-se o conjunto B, um conjunto contendo as sentenças representando os ramos abertos de A (linha 3). Após a formação do conjunto representando os ramos abertos tenta-se formular um conjunto de hipóteses que contenha proposições que fechem todos os ramos de B (linha 4). Se não for possível formular este conjunto de hipóteses então varre-se o conjunto B procurando por conjuntos de hipóteses que fechem pelo menos um ramo de B (linhas 5 a 10). Após a formulação destas hipóteses, estas são combinadas uma a uma para formular um conjunto que feche todos os ramos de B (linha 11).

**Exemplo II.2:** Considerando os conjuntos  $T = \{ p \rightarrow q \}$  e  $F = \{ q \}$ , constrói-se o tableau referente à  $T \cup \neg F$  Figura II.5.



**Figura II.5:** Construção do tableau  $p \rightarrow q \wedge \neg q$ .

O conjunto  $B$  é preenchido com os ramos abertos do tableau, neste exemplo  $B = \{ \neg p \wedge \neg q \}$ . Aqui o conjunto  $H$  é preenchido com todas as proposições que venham a fechar os ramos de  $B$ , tais proposições não podem ser iguais ao fato presente em  $F$ , neste exemplo a única proposição que corresponde ao fechando do ramo apresentado em  $B$  é  $p$ , sendo assim  $H = \{ p \}$ .

## II.4 – Programação Lógica Abdutiva (ALP)

A visão de Programação Lógica Abdutiva apresentada nessa seção é baseada no trabalho de (Denecker & Kakas, 2002).

A realização de raciocínios abdutivos na ALP possuem semelhança com consultas em Prolog. Na ALP precisa-se de um programa lógico  $T$ , ou uma base de conhecimento em Prolog, um conjunto de predicados abdutíveis (potenciais hipóteses)  $A$ , um conjunto de restrições de integridade  $IC$  e uma consulta  $F$ , tal como consultas em Prolog. Nestes conjuntos é aplicado o mecanismo de resolução para, no caso da ALP, encontrar em  $A$  um conjunto de hipóteses  $H$  tal que  $H$  não seja conflitante com os elementos de  $IC$ , e que juntamente com a teoria  $T$  explique os fatos  $F$ , em Prolog as busca-se soluções para as consultas na base de conhecimento.

O Exemplo II.3 apresenta o funcionamento de um raciocínio abduativo na ALP.

**Exemplo II.3.** Considerando como exemplo um circuito elétrico constituído de duas lâmpadas e três interruptores temos:

**Um programa lógico (T):**

- $lampada\_1\_acesa \leftarrow interruptor\_1\_acionado.$
- $lampada\_2\_acesa \leftarrow interruptor\_2\_acionado.$
- $lampada\_2\_acesa \leftarrow interruptor\_3\_acionado.$
- $interruptor\_2\_quebrado.$

**Predicados Abdutíveis (A):**

- $interruptor\_1\_acionado.$
- $interruptor\_2\_acionado.$
- $interruptor\_3\_acionado.$

**Restrições de integridade (IC):**

- $Falso \leftarrow interruptor\_2\_acionado \wedge interruptor\_2\_quebrado.$

**Consulta (F):**

- $lampada\_1\_acesa, lampada\_2\_acesa.$

Em um sistema que implemente a ALP, ao entrar com os fatos *lampada\_1\_acesa* e *lampada\_2\_acesa* em forma de consulta, o sistema utiliza a o programa lógico (teoria), os predicados abdutíveis (A) e as restrições de integridade (IC) para formular um conjunto H de hipóteses que explicam os fatos (F) da consulta. Neste exemplo a hipótese formulada é *interruptor\_1\_acionado*  $\wedge$  *interruptor\_3\_acionado*, pois a hipótese *interruptor\_2\_acionado* é conflitante com as restrições de integridade e o programa lógico, que indicam que se o interruptor 2 estiver quebrado e ligado, e ele estando quebrando segundo o programa lógico, então a lâmpada 2 não se acenderá por ele.

# Apêndice III

## Questionário

Faculdade Campo Limpo Paulista

Programa de Mestrado em Ciência da Computação

### *Perfil*

**Idade:** \_\_\_\_\_ **Sexo:** F  M  **Escolaridade:** Fundamental  Médio   
Graduação  Pós-graduação

### *Breve Instrução*

Este questionário envolve perguntas sobre situações simples que encontramos no dia a dia. Ele faz parte de uma pesquisa sobre como encontramos uma possível resposta quando não dispomos de uma resposta exata. Ou seja, nas situações propostas, escolha a resposta que você suspeita que melhor explica a situação. Por exemplo:

### *Situação exemplo*

José possui na frente de sua casa um grande gramado. Um dia José chega à sua casa e observa que **o gramado está molhado**. Considerando somente que:

- **Irrigadores** instalados podem molhar o gramado.
- **Chuva** pode molhar o gramado.

Escolha a alternativa que melhor explique o gramado de José estar molhado aquele dia.

- a) **Irrigadores** estiveram ligados antes da chegada de José.
  - b) **Choveu** antes da chegada de José.
  - c) **Choveu** e os **irrigadores** estiveram ligados antes da chegada de José.
  - d) Ocorreu de **chover** ou os **irrigadores** estiveram ligados antes da chegada de José.
-

### *Situação 1*

Novamente José chega e depara-se com **o gramado molhado**. Considerando somente que:

- **Irrigadores** instalados podem molhar o gramado.
- **Chuva** pode molhar o gramado.
- No **inverno** não chove na região onde José vive.

Sabendo-se que é **inverno** na região onde José mora, escolha a alternativa que melhor explique **o gramado de José estar molhado** naquele dia:

- a) **Irrigadores** estiveram ligados antes da chegada de José.
- b) **Choveu** antes da chegada de José.
- c) **Choveu** e os **irrigadores** estiveram ligados antes da chegada de José.
- d) Ocorreu de **chover** ou os **irrigadores** estiveram ligados antes da chegada de José.

## Situação 2

Doenças são manifestadas por sintomas. Abaixo está uma relação de doenças e seus respectivos sintomas.

Doença	Sintomas
Pneumonia	Apenas <b>febre, coriza e dor de cabeça</b>
Resfriado	Apenas <b>febre e coriza</b>
Rinite Alérgica	Apenas <b>dor de cabeça e coriza</b>

O Sr. Márcio está apresentando um quadro sintomático composto por: **febre, coriza e dor de cabeça**. Considerando apenas as informações apresentadas na tabela acima, você conseguiria explicar os sintomas do Sr. Márcio?

- a) Estou convicto de que é **pneumonia**.
- b) Estou convicto de que é **resfriado**.
- c) Estou convicto de que é **rinite alérgica**.
- d) Existe um quadro simultâneo de **resfriado** e também de **rinite alérgica**.
- e) Existe um quadro simultâneo de **pneumonia**, de **resfriado** e também de **rinite alérgica**.

### Situação 3

Considere a relação entre doenças e sintomas a seguir:

Doença	Sintomas
Gripe	Apenas <b>febre, dor de cabeça, dor muscular e espirros</b>
Dengue	Apenas <b>febre, dor de cabeça, dor muscular e manchas vermelhas</b>

O filho de Alberto encontra-se no médico com o seguinte quadro de sintomas: **febre, dor de cabeça, dor muscular e falta de ar**. Utilizando somente os conhecimentos da tabela acima, você conseguiria explicar os sintomas apresentados pelo filho de Alberto?

- a) Estou convicto de que é **gripe**.
- b) Estou convicto de que é **dengue**.
- c) Existe um quadro simultâneo de **gripe** e também **dengue**.
- d) Poderia ser **gripe** ou **dengue**.

#### *Situação 4*

Considere a relação entre doenças e sintomas a seguir:

<b>Doenças</b>	<b>Sintomas</b>
Bronquite	Apenas <b>tosse e febre</b>
Asma	Apenas <b>tosse</b>
Laringite	Apenas <b>garganta inflamada</b>

Novamente o filho de Alberto encontra-se no médico desta vez ele apresenta um quando de sintomas com: **febre, arrepios, dor de cabeça, dor muscular, garganta inflamada, nariz entupido e tosse**. Baseado somente na tabela acima você conseguiria explicar os sintomas apresentados pelo filho de Alberto?

- a) Estou convicto de que é **bronquite**.
- b) Estou convicto de que é **asma**.
- c) Estou convicto de que é **laringite**.
- d) Existe um quadro simultâneo de **bronquite e laringite**.
- e) **Não estou convicto** sobre qual(s) doença(s) apresenta o filho de Alberto.

### *Situação 5*

No Bairro 27 houve um crime. José foi morto a facadas. Pedro é réu em um julgamento no qual é acusado de matar José, pois **suas digitais foram encontradas na arma do crime**. É possível que tenha acontecido uma das coisas:

- 1- **Pedro usou a arma do crime para matar José**, por isso as digitais são encontradas na arma do crime.
- 2- **Pedro socorreu José retirando a faca do abdômen dele**, por isso as suas digitais são encontradas na arma do crime.

Além disso:

- Se **Pedro socorreu José** então **Pedro é inocente**.
- Se **Pedro matou José** então **Pedro é culpado**.

Colocando-se no papel de um **advogado de acusação**, cuja função é **incriminar um réu**, como você explicaria o que ocorreu no Bairro 27?

- a) Pedro usou a arma do crime para matar José.
- b) Pedro socorreu José retirando a arma do crime de seu corpo.
- c) Pedro matou José.
- d) Pedro matou José e depois tentou socorrê-lo.

### *Situação 6*

Considere que o trajeto entre Firenze e Siena pode ser realizado:

- Por trem.
- Por automóvel.
- À cavalo.
- Por avião.

No ano de 1470, Leonardo da Vinci saiu de Firenze, onde vivia, e viajou até Siena para visitar uma tia. Ele utilizou nesta viagem:

- a) Um trem.
- b) Um automóvel.
- c) Um cavalo.
- d) Um avião.

### *Situação 7*

Josué encontra-se no deserto e observa algo verde, distante. O que melhor explica o que Josué está vendo?

- a) Estou convicto de que é um gramado.
- b) Estou convicto de que é um cacto.
- c) Estou convicto de que é uma bandeira verde.
- d) Poderia ser um cacto ou uma bandeira verde.

### *Situação 8*

Crimes são classificados de acordo com os atos criminais realizados pelo criminoso. Abaixo está uma relação entre crimes e os atos que os classificam.

- É classificado como **roubo** o ato de tomar uma propriedade de alguém.
- É classificado como **latrocínio** o ato de tomar uma propriedade de alguém e executar a vítima.
- É classificado como **homicídio** o ato de executar a vítima.

O Sr. Geraldo registrou um boletim de ocorrência no qual a sua irmã Carla teve o seu **celular tomado** por dois bandidos e em seguida foi **esfaqueada até a morte**. Considerando apenas as relações apresentadas acima, escolha a alternativa que melhor explique os atos criminais registrados:

- Roubo.
- Homicídio.
- Latrocínio.
- Roubo e homicídio.
- Latrocínio e roubo.
- Latrocínio e homicídio.

### *Situação 9*

O detetive William está analisando a cena do crime do homicídio da Rua 34. Na experiência do detetive William ele aprendeu que:

- Se uma **vítima for ferida**, haverá sangue no local.
- Se **houver disparo de arma** serão encontrados cartuchos vazios no local.
- **Homicídios são classificados como culposos** se na cena do crime há sinais de confronto.

No crime da Rua 34 o detetive William encontrou seguintes evidências: **Há sangue, um cartucho vazio no chão e há sinais de confronto**. Baseado nos conhecimentos do detetive William, escolha a alternativa que melhor explica as evidências encontradas.

- a) A **vítima foi ferida**.
- b) Uma **arma que foi disparada**.
- c) Ocorreu um **homicídio culposo**.
- d) Houve um **homicídio culposo** em que a **vítima foi ferida** por um **disparo de arma**.

### *Situação 10*

O Sr. Mario encontra com o Sr. Antônio na rua e comenta com ele sobre o amigo Luiz dizendo: “Luiz entrou na casa rapidamente.”. Qual interpretação o Sr. Antônio teria dessa frase?

- a) Luiz recolheu-se ao seu lar com pressa.
- b) Luiz invadiu a casa.
- c) A frase tem um **substantivo próprio** “Luiz”, um **verbo** “entrou”, um **substantivo comum** “casa” e um **advérbio** “rapidamente”.

**Gabarito do Questionário (alternativas computadas pelo algoritmo Peirce).**

<i>Situação</i>	<i>Alternativa</i>
1	a
2	a
3	d
4	e
5	a
6	c
7	d
8	c
9	d
10	a

# Apêndice IV

## Pseudocódigo do compilador da linguagem Peirce Abdutiva

**Algoritmo** Programa()

```
{  
    retornar proposicoesDeclarativas();  
};
```

**Algoritmo** proposicoesDeclarativas()

```
{  
    proposicoes := dclTeoria();  
    proposicoes += dclCondicao();  
    proposicoes += dclFatos();  
    retornar proposicoes;  
}
```

**Algoritmo** dclTeoria()

```
{  
    token := anaLex.proximoToken();  
    se token <> TOKEN_TEORIA então  
        Erro ("Theory esperado.");  
    senão  
    {  
        token := anaLex.proximoToken();  
        se token <> TOKEN_IDENTIFICADOR então  
            Erro ("Identificador esperado");  
        senão  
        {  
            token := anaLex.proximoToken();  
            se token <> "{" então  
                Erro ("{" esperado.");  
            senão  
            {  
                proposicao := "{" + sequenciaProposicoesHF();  
                token := anaLex.proximoToken();  
                se token <> "}" então  
                    Erro ("}" esperado.");  
                senão  
                {  
                    token := anaLex.proximoToken();  
                    se token <> ";" então  
                        Erro (";" esperado.");  
                    senão  
                    {  
                        proposicao := proposicao + " ";  
                        retornar proposicao;  
                    }  
                }  
            }  
        }  
    }  
}
```

```

    }
  }
}

```

**Algoritmo** sequenciaProposicoesHF()

```

{
  proposicao := proposicaoHF();
  token := anaLex.peek0();
  enquanto token = "," então
  {
    token := anaLex.peek0();
    proposicao := proposicao + "," + ();
    token := anaLex.peek0();
  }
  retornar proposicao;
}

```

**Algoritmo** proposicaoHF()

```

{
  se (anaLex.peek1() <> TOKEN_E) e (anaLex.peek2() <> TOKEN_E) então
  {
    se (anaLex.peek1() <> TOKEN_OU) e (anaLex.peek2() <> TOKEN_OU) então
    {
      se (anaLex.peek1() <> TOKEN_IMPLICACAO) e (anaLex.peek2() <>
        TOKEN_IMPLICACAO) então
      {
        //Literal
        retornar proposicaoConjuntiva();
      }
      senão
      {
        se anaLex.peek0() <> TOKEN_NEGACAO então
        {
          se anaLex2() <> TOKEN_NEGACAO então
          {
            //Forma 3: a => + (...)
            retornar proposicaoConjuntiva();
          }
          senão
          {
            se anaLex.peek4() <> TOKEN_OU então
            {
              //Forma 3: a => !b ^ (...)
              retornar proposicaoConjuntiva();
            }
            senão
            {
              //Forma 4: a => !b v !(...)
              retornar proposicaoDisjuntiva();
            }
          }
        }
      }
    }
    senão
    {
      //Forma 4: ¬a => !(...)
      retornar proposicaoDisjuntiva();
    }
  }
}

```

```

    }
    senão
        retornar proposicaoDisjuntiva();
    }
    senão
        retornar proposicaoConjuntiva();
}

```

**Algoritmo** proposicaoConjuntiva()

```

{
    proposicaoEsquerda := sequenciaProposicoesComConjuncao();
    token := anaLex.peek0();
    se token <> TOKEN_IMPLICACAO então
    {
        se há átomos negados em proposicaoEsquerda então
            Erro ("Átomos do lado esquerda da proposição esperados.");
        senão
        {
            token := anaLex.proximoToken();
            proposicaoEsquerda := NegarProposicoes(proposicaoEsquerda);
            proposicaoDireita := sequenciaProposicoesComConjuncao();

            listaProposicoesDireita := separar(proposicaoDireita, ",");

            proposicao := "{" + proposicaoEsquerda + "," +
                listaProposicoesDireita[1] + "}";

            para i := 2 até tamanho(listaProposicoesDireita) faça
            {
                proposicao := proposicao + "," + "{" + proposicaoEsquerda +
                    "," + listaProposicoesDireita[i] + "}";
            }

            //Forma 3
            retornar proposicao;
        }
    }
    senão
    {
        se anaLex.peek0 <> AnaLex.TOKEN_IMPLICACAO
        {
            //Forma 1
            retornar proposicaoEsquerda;
        }
        senão
        {
            Erro (Erro Forma HF 3: Átomos esperados à esquerda do conectivo
                lógico de implicação);
        }
    }
}

```

**Algoritmo** sequenciaProposicoesComConjuncao()

```

{
    token := anaLex.proximoToken();
    se token <> Proposicao então
        Erro ("Proposição esperada.");
    senão
    {

```

```

proposicao := "{" + identificarProposicao();
token := anaLex.peek0();
se token <> TOKEN_E então
{
    se token <> TOKEN_OU então
    {
        retornar proposicao + "}";
    }
    senão
    {
        Erro ("Conectivo lógico de conjunção esperado.");
    }
}
senão
{
    token := anaLex.proximoToken();
    proposicao := proposicao + "}" + "," +
        sequenciaProposicoesComConjuncao();
    retornar proposicao;
}
}
}

```

**Algoritmo** proposicaoDisjuntiva()

```

{
    proposicaoEsquerda := "{" + sequenciaAtomosNegadosComDisjuncao() +
    "}";
    token := anaLex.peek0();
    se token <> TOKEN_IMPLICACAO então
    {
        se não há átomos em proposicaoEsquerda então
        {
            //Forma 2
            retornar proposicaoEsquerda;
        }
        senão
        {
            Erro ("Átomos negados esperado.");
        }
    }
    senão
    {
        token := anaLex.proximoToken();
        proposicaoEsquerda = NegarÁtomosERetirarNegaçãoDeÁtomosNegados(
            proposicaoEsquerda);
        listaProposicoesEsquerda := separar(proposicaoEsquerda, ",");

        proposicaoDireita := "{" + sequenciaAtomosNegadosComDisjuncao() +
        "}";

        se não há átomos em proposicaoDireita então
        {
            proposicao := "{" + listaProposicoesEsquerda[0] + "," +
            proposicaoDireita + "}";

            para i := 1 até tamanho(listaProposicoesEsquerda) faça
            {
                Proposicao := proposicao + "," + "{" +
                listaProposicoesEsquerda[i] + "," + proposicaoDireita + "}";
            }
        }
    }
}

```

```

    }

    //Forma 4
    retornar sentenca;
  }
  senão
  {
    Erro ("Átomos negados do lado direito da proposição esperados.")
  }
}
}

```

**Algoritmo** sequenciaProposicoesComDisjuncao()

```

{
  token := anaLex.proximoToken();
  se token <> Proposicao então
    Erro ("Proposição esperada.");
  senão
  {
    proposicao := identificarProposicao(token);
    token := anaLex.peek0();
    se token <> TOKEN_OU então
    {
      se token <> TOKEN_E então
      {
        retornar proposicao;
      }
      senão
      {
        Erro ("Conectivo lógico de disjunção esperado.");
      }
    }
    senão
    {
      token := anaLex.proximoToken();
      proposicao := proposicao + "," +
        sequenciaProposicoesComDisjuncao();
      retornar proposicao;
    }
  }
}

```

**Algoritmo** identificarProposicao(token)

```

{
  se token <> TOKEN_NAO então
  {
    se token <> TOKEN_IDENTIFICADOR então
    {
      Erro ("Proposição esperada.");
    }
    senão
    {
      retornar token.ID;
    }
  }
  senão
  {
    token := anaLex.proximoToken();
    se token <> TOKEN_IDENTIFICADOR então

```

```

    {
        Erro ("Identificador após símbolo de negação esperado");
    }
    senão
    {
        retornar "¬" + token.ID;
    }
}
}

```

**Algoritmo** dclCondicoes()

```

{
    token := anaLex.proximoToken();
    se token <> "Conditions" então
        Erro ("Conditions esperado.");
    senão
    {
        token := anaLex.proximoToken();
        se token <> TOKEN_IDENTIFICADOR então
            Erro ("Identificador esperado");
        senão
        {
            token := anaLex.proximoToken();
            se token <> "{" então
                Erro ("{ esperado.");
            senão
            {
                token := anaLex.peek0();
                se token <> "}" então
                {
                    proposicao := "{" + sequenciaProposicoesHF();
                    token := anaLex.proximoToken();
                    se token <> "}" então
                        Erro ("} esperado.");
                    senão
                    {
                        token := anaLex.proximoToken();
                        se token <> ";" então
                            Erro ("; esperado.");
                        senão
                        {
                            proposicao := proposicao + " ";
                            retornar proposicao;
                        }
                    }
                }
            }
        }
    }
    senão
    {
        retornar "";
    }
}
}
}

```

**Algoritmo** dclFacts()

```

{
    token := anaLex.proximoToken();
    se token <> "Facts" então

```

```

    Erro ("Facts esperado.");
senão
{
    token := anaLex.proximoToken();
    se token <> TOKEN_IDENTIFICADOR então
        Erro ("Identificador esperado");
    senão
    {
        token := anaLex.proximoToken();
        se token <> "{" então
            Erro ("{ esperado.");
        senão
        {
            proposicao := "{" + sequenciaAtomosFatos();
            token := anaLex.proximoToken();
            se token <> "}" então
                Erro ("} esperado.");
            senão
            {
                token := anaLex.proximoToken();
                se token <> ";" então
                    Erro ("; esperado.");
                senão
                {
                    proposicao := proposicao + ";";
                    retornar proposicao;
                }
            }
        }
    }
}

```

**Algoritmo** sequenciaAtomosFatos()

```

{
    token := anaLex.proximoToken();
    se token <> Átomo então
        Erro ("Átomo esperado.");
    senão
    {
        proposicao := "{" + identificarAtomo(token);
        token := anaLex.peek0();
        se token <> TOKEN_E então
        {
            se token <> TOKEN_OU então
            {
                retornar proposicao + "{";
            }
            senão
            {
                Erro ("Conectivo lógico de conjunção esperado.");
            }
        }
    }
    senão
    {
        token := anaLex.proximoToken();
        proposicao := proposicao + "}" + "," + sequenciaAtomosFatos();
        retornar proposicao;
    }
}

```

```
    }  
  }  
  
  Algoritmo identificarAtomo(token)  
  {  
    se token <> TOKEN_IDENTIFICADOR então  
    {  
      Erro ("Átomo esperado.");  
    }  
    senão  
    {  
      retornar "+" + token.ID;  
    }  
  }  
}
```

# Apêndice V

## Artigo Publicado

O artigo *Peirce - an Algorithm for Abductive Reasoning Operating with a Quaternary Reasoning Framework*, foi submetido e aceito na *13<sup>th</sup> Mexican International Conference on Artificial Intelligence (MICAI)*, *Qualis-CAPES B2 em Ciência da Computação*. O artigo foi publicado nos anais da conferência e também no *Research in Computer Science Journal*, *ISSN 1870-4069*.

# Peirce: an Algorithm for Abductive Reasoning Operating with a Quaternary Reasoning Framework

Felipe Rodrigues, Carlos Eduardo A. Oliveira, and Osvaldo Luiz de Oliveira

FACCAMP, Campo Limpo Paulista (SP),  
Brazil

rodrigues\_felipe7@hotmail.com, carlos.br@gmail.com, osvaldo@faccamp.br

**Abstract.** Abductive reasoning algorithms formulate possible hypotheses to explain observed facts using a theory as the basis. These algorithms have been applied to various domains such as diagnosis, planning and interpretation. In general, algorithms for abductive reasoning based on logic present the following disadvantages: (1) they do not allow the explicit declaration of conditions that may affect the reasoning, such as intention, context and belief; (2) they allow little or no consideration for criteria required to select good hypotheses. Using Propositional Logic as its foundation, this study proposes the algorithm Peirce, which operates with a framework that allows one to explicitly include conditions to conduct abductive reasoning and uses a criterion to select good hypotheses that employs metrics to define the explanatory power and complexity of the hypotheses. Experimental results suggest that abductive reasoning performed by humans has the tendency to coincide with the solutions computed by the algorithm Peirce.

**Keywords:** Abductive reasoning, automated Reasoning, logic, human factors.

## 1 Introduction

Abductive reasoning formulates hypotheses to explain observed facts using a theory as the basis. Numerous intellectual tasks make use of abductive reasoning, including medical diagnostics, fault diagnostics, scientific discovery, legal argumentation and interpretation.

Abductive reasoning algorithms based on logic frequently operate with the Theory, Hypotheses and Facts (THF) ternary reasoning framework (as shown in [2], [4], [5] and [11]). When these algorithms are formalized using Propositional Logic (PL) [9], the THF framework is frequently instantiated according to Definition 1.

**Definition 1 (THF reasoning framework).** The THF reasoning framework for abductive reasoning is a system  $\langle T, H, F \rangle$  that consists of the following:

- A finite and non-empty theory set,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , of PL sentences denoting  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ . This set represents the hypotheses that must be assumed as truth during the reasoning process.

- A finite hypotheses set,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , of PL sentences denoting  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . This set represents the hypotheses that along with the set T explain the facts represented by the set F.
- A set with a single fact,  $F = \{ f \}$ , where  $f$  is a PL literal (atom or negated atom). This set represents an occurrence of an evidence, a manifestation, a symptom, an observation, a mark or a sign to be explained through abductive reasoning.

Having the T and F sets as input, an abductive reasoning algorithm should find a set of hypotheses H that satisfies the following conditions:

$$T \not\models F, \quad (1)$$

$$T \cup \{ h \} \models F, \forall h \in H, \quad (2)$$

$$T \cup \{ h \} \not\models \perp, \forall h \in H, \quad (3)$$

$$\{ h \} \not\models F, \forall h \in H. \quad (4)$$

The statements above refer to the concept of logical consequence described in Definition 2.

**Definition 2 (A  $\models$  B, i.e., B is a logical consequence of A).** Let  $A = \{ a_1, a_2, a_3, \dots, a_n \}$  and  $B = \{ b_1, b_2, b_3, \dots, b_m \}$  be two finite and non-empty sets of PL sentences. Then,  $A \models B$  if and only if the interpretations in which  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n$  is true,  $b_1 \wedge b_2 \wedge b_3 \wedge \dots \wedge b_m$  is also true.

The condition (1) prevents that the theory set T alone has as logical consequence the facts set F. Hypotheses satisfying condition (2) are called **candidate hypotheses**, and they can explain the single fact denoted by F. Candidate hypotheses satisfying condition (3) are called **consistent hypotheses**. Conversely, candidate hypotheses that do not satisfy condition (3) are called inconsistent hypotheses and should be discarded. Candidate hypotheses that satisfy condition (4) are called **explanatory hypotheses**. Conversely, candidate hypotheses that do not satisfy condition (4) are called non-explanatory hypotheses and should be discarded.

**Example 1.** Joseph has a large lawn in front of his house. One day, Joseph arrives at home and observes that the lawn is wet. Considering only that (1) rain can make the lawn wet and that (2) sprinklers installed across the lawn can make it wet, which hypotheses can explain the fact that the lawn is wet?

One possible formalization using the THF framework consists in defining:

- Propositions ‘ $r$ : **Rain** occurred’, ‘ $s$ : **Sprinklers** were activated’ and ‘ $w$ : Lawn is **wet**’.
- A theory set  $T = \{ r \rightarrow w, s \rightarrow w \}$ .
- A fact set  $F = \{ w \}$ .

The theory set T and the fact set F satisfy condition (1), whereas a theory  $T_1 = \{ r \rightarrow w, s \rightarrow w, w \}$  has F as its logical consequence; therefore  $T_1$  and F do not satisfy condition (1). Let  $H = \{ r, s, r \wedge s, r \wedge \neg w, w \}$  be a set of candidate hypotheses. Each hypothesis  $h \in H$  satisfies condition (2), and each hypothesis  $h \in \{ r, s, r \wedge s, w \}$  satisfies condition (3), i.e., they are consistent. However, the hypothesis  $r \wedge \neg w$  is inconsistent because  $T \cup \{ r \wedge \neg w \} \models \perp$ ; therefore, it must be discarded. Each

hypothesis  $h \in \{ r, s, r \wedge s \}$  satisfies condition (4); however, the hypothesis  $w$  is not explanatory because  $\{ w \} \models F$ ; therefore, it must also be discarded. Thus, removing the inconsistent and non-explanatory hypotheses from  $H$ , we obtain  $H = \{ r, s, r \wedge s \}$ .

In general, abductive algorithms work as follows: having the theory set  $T$  and the fact set  $F$  as the input, the algorithm verifies whether or not the condition (1) has been satisfied; if the condition (1) is not satisfied, then there are no hypotheses to be formulated because  $F$  is a logical consequence of  $T$ ; however, if the condition (1) is satisfied, then the algorithm formulates a finite set of possible hypotheses  $H$  that satisfies the condition (2). Next, the algorithm removes from  $H$  the hypotheses that do not satisfy the conditions (3) and (4), and thus returning the resulting set  $H$  as an answer by the algorithm.

Some algorithms, however, include an additional step with the goal of letting in  $H$  only hypotheses considered good, according to extra-logical criteria. A criterion commonly used is “simplicity”, which considers, for example, an atomic hypothesis better than a composite hypothesis, e.g.,  $r$  is better than  $r \wedge s$ .

The need to represent conditions such as context, circumstance and intention is common and important when conducting abductive reasoning. For example, reasoning to make a medical diagnosis considering the context of diseases of a region. Operating with a THF reasoning framework, the existing algorithms to perform abductive reasoning have the disadvantage of forcing the representation of these conditions in the theory set  $T$ . This solution is not appropriate because representing conditions in the theory set  $T$  mischaracterizes the theory, making it less general and more *ad hoc* (specific to explain what one wants to explain).

Abductive reasoning formulates hypotheses, and some of these hypotheses may be better at explaining the facts than others. Today we do not know, exactly, which criteria determine what makes a hypothesis better than another, authors from several fields [3] [8] [10] [13] [16] [17] have suggested that abductive reasoning involves the selection of good hypotheses. However, the existing abductive reasoning algorithms have the disadvantage of dedicating little or no consideration for criteria required to select good hypotheses.

Many practical applications of reasoning require the definition of a set of  $n \geq 2$  facts. However, the many existing algorithms have the disadvantage of operating with only a single fact.

This work proposes an algorithm, called Peirce, that performs abductive reasoning, and this algorithm differs from the existing solutions mainly because (1) it works with a reasoning framework called TCHF (Theory, accepted Conditions, Hypotheses and Facts), thus allowing conditions to be explicitly represented; (2) it allows  $n \geq 2$  facts to be represented; and (3) it introduces a criterion to select good hypotheses that employ metrics to define the explanatory power and the complexity of the hypotheses.

Section 2 describes the algorithm Peirce, dedicating particular attention to the design and operation of the TCHF reasoning framework (Subsection 2.1) and the definition of a criterion to select good abductive hypotheses (Subsection 2.2). The pseudocode for the algorithm Peirce is presented and discussed in Subsection 2.3. Section 3 details an experimental study conducted to verify whether the solutions computed by the algorithm Peirce tend to coincide with the abductive reasoning

performed by humans. Section 4 describes related works, highlighting the differences with this work. Section 5 presents the conclusions.

## 2 The Algorithm Peirce

The abductive reasoning algorithm proposed in this study has been named Peirce in honor of the American philosopher Charles Sanders Peirce, who created the concept of abductive reasoning [14]. The following subsections detail the reasoning framework used by the algorithm Peirce, a criteria to select good hypotheses and the pseudocode of the algorithm.

### 2.1 TCHF Reasoning Framework

The TCHF reasoning framework proposed in this study differs from the classic THF reasoning framework (Definition 1) by including the accepted conditions set C and by redefining the facts set F to allow the declaration of not just one single fact, but rather a finite number of one or more facts. The TCHF framework is formalized in Definition 3 and uses the PL sentences in HF form as specified in Definition 4.

**Definition 3 (TCHF reasoning framework).** The TCHF framework for abductive reasoning is a system  $\langle T, C, H, F \rangle$  consisting of the following:

- A finite and non-empty theory set,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , of PL sentences in HF form denoting  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ . This set represents the hypotheses that must be assumed as truth during the reasoning process.
- A finite hypotheses set,  $H = \{ h_1, h_2, h_3, \dots, h_n \}$ , of PL sentences in HF form denoting  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_n$ . This set represents the hypotheses that along with the sets T and C explain the facts represented by the set F.
- A finite accepted conditions set,  $C = \{ c_1, c_2, c_3, \dots, c_p \}$ , of PL sentences in HF form denoting  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_p$ . This set represents the conditions that must be assumed as truth during the reasoning process.
- A finite and non-empty facts set  $F = \{ f_1, f_2, f_3, \dots, f_q \}$  of PL positive literals, denoting  $f_1 \wedge f_2 \wedge f_3 \wedge \dots \wedge f_q$ . The role of this set is to represent evidences, manifestations, symptoms, observations, marks or signs to be explained by the abductive reasoning.

**Definition 4 (HF form).** A sentence of PL in the HF form is an acyclic sentence written in one of the following formats:

- $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n$ , where  $a_i$  ( $1 \leq i \leq n$ ) are literals.
- $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n$ , where  $a_i$  ( $1 \leq i \leq n$ ) are negative literals.
- $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n \rightarrow b_1 \wedge b_2 \wedge b_3 \wedge \dots \wedge b_m$ , where  $a_i$  ( $1 \leq i \leq n$ ) are positive literals and  $b_j$  ( $1 \leq j \leq m$ ) are literals.
- $a_1 \vee a_2 \vee a_3 \vee \dots \vee a_n \rightarrow b_1 \vee b_2 \vee b_3 \vee \dots \vee b_m$ , where  $a_i$  ( $1 \leq i \leq n$ ) are literals and  $b_j$  ( $1 \leq j \leq m$ ) are negative literals.

The restriction of TCHF framework to sentences in the HF form aims to make the algorithm Peirce run the conversion of sentences in polynomial time, because sentences in HF form can be easily converted into Horn Clauses [9] (disjunction of literals with at most one positive literal). As will be described in Subsection 2.3, algorithm Peirce uses Resolution as the inference mechanism and this mechanism can be efficiently implemented on Conjunctive Normal Form (CNF) sentences with Horn Clauses.

The set C gives the TCHF reasoning framework the advantage of allowing the explicit definition of conditions that in the classical THF framework, would normally be declared within the theory set T. Thus, the set C avoids “contaminating” the set T with sentences that fundamentally do not belong to the theory. Moreover, this makes it easier to represent two or more instances of abductive reasoning that share the declarations of T and F but differ in the set of accepted conditions. Example 2, which is described next, illustrates the use of the TCHF reasoning framework.

**Example 2.** Consider once more the scenario described in Example 1 in which Joseph arrives at home and observes his lawn wet. However, let us say that Joseph knows that the water tank supplying the sprinklers has been empty for a month; therefore, under this condition, the sprinklers could not have been activated.

One possible formalization using the TCHF framework is to define the following:

- Propositions ‘*r*: **Rain** occurred’, ‘*s*: **Sprinklers** were activated’, ‘*w*: Lawn is **wet**’ and ‘*t*: Water **tank** that supplies sprinklers is empty’.
- A theory  $T = \{ r \rightarrow w, s \rightarrow w \}$ , which is the same as in Example 1.
- A set of accepted conditions  $C = \{ t, t \rightarrow \neg s \}$ .
- A set of facts  $F = \{ w \}$ , which is the same as in Example 1.

The conditions in abductive reasoning are motivated by several factors which are linked to context (information associated to space), circumstances (information associated with time), intention (manifestation of the will to reach some wanted conclusions), belief or faith (information that is accepted on principle) etc. Examples of specific conditions used in abductive reasoning are as follows: (1) In abductive reasoning used for medical diagnoses, regional context may allow one to specify a set of diseases that are common or uncommon for a given region; (2) In abductive reasoning used for anthropological studies, the specification of possible agents that might have been responsible for the death of a hominid based on the knowledge that the hominid lived 4 million years ago (circumstance); (3) In abductive reasoning for judicial decisions, possible conditions may be specified with the intent of acquitting (or condemning) a defendant; and (4) In abductive reasoning for religious or metaphysical beliefs, the faith or belief that there is life after death can be declared as a condition upon which reasoning are made.

Taking the sets T and F as inputs, an abductive reasoning algorithm operating with the TCHF framework should find a set of hypotheses H that satisfies the following conditions:

$$T \cup C \neq F, \tag{5}$$

$$T \cup C \cup \{ h \} \models_p F, \forall h \in H, \tag{6}$$

$$T \cup C \cup \{ h \} \neq \perp, \forall h \in H, \quad (7)$$

The condition (6) uses the partial logical consequence as defined in Definition 5.

**Definition 5 (A  $\models_p$  B, i.e., B is a partial logical consequence of A).** Let  $A = \{ a_1, a_2, a_3, \dots, a_n \}$ ,  $B = \{ b_1, b_2, b_3, \dots, b_m \}$  and  $C = \{ c_1, c_2, c_3, \dots, c_q \}$ ,  $C \subseteq B$ , be three finite and non-empty sets of PL sentences. Then,  $A \models_p B$  if only if the interpretations in which  $a_1 \wedge a_2 \wedge a_3 \wedge \dots \wedge a_n$  is true,  $c_1 \wedge c_2 \wedge c_3 \wedge \dots \wedge c_q$  is also true.

## 2.2 Selection of Good Abductive Hypotheses

In general, several hypotheses may be able to explain observed facts. However, certain hypotheses may explain facts better than others. Therefore, abductive reasoning can be observed as a process that formulates  $m \geq 1$  general hypotheses followed by the selection of  $n \leq m$  good hypotheses. Naturally, selection criteria must be established, but it is still difficult to define the conditions that make a hypothesis good.

Contemporary philosophers have analyzed the issue of selecting good hypotheses. Harman [8] considers abduction to be an inference of the best explanation and argues that the best hypothesis is the simplest, most plausible and is the least *ad hoc*. By comparing theories (e.g., Darwin's Theory of Evolution vs. Creationist Theory or Lavoisier's Theory of Combustion vs. Phlogiston Theory), Thagard [16, 17] establish criteria that explain the preference for one hypothesis over another and considers the best hypothesis to be the most consilient (explains more facts), the most simple, and it would provide the best analogy with hypotheses that explain facts in other domains.

Criteria to select good hypotheses have been extensively studied in the fields of philosophy (e.g., [2] [8] [16]), psychology (e.g., [13]) and artificial intelligence (e.g., [3] [10] [15]). However, the precise formulation of these criteria remains controversial. In general, factors such as the "explanatory power" and the "complexity" of a hypothesis are recurrent and have similar connotations across several studies. Therefore, this study has proposed using these two factors to develop a selection criterion. Aiming at the development of algorithms to perform abduction that need dealing with quantitative measures for the explanatory power and the complexity of a hypothesis, this study proposes an understanding of these factors as follows:

- Explanatory power (or comprehensiveness): the explanatory power of a hypothesis quantifies the degree to which it is capable of explaining the facts involved in the reasoning. A metric for a hypothesis' explanatory power is given by the ratio between the number of facts it can explain and the total number of facts to be explained by the abductive reasoning process. For example, a hypothesis that explains 4 out of 5 facts has an explanatory power of 4/5, and a hypothesis that explains all of the facts has an explanatory power of 1.
- Complexity: the complexity factor refers to how many different elements and relationships are present in a hypothesis. A metric for hypothesis complexity is the number of atomic propositions that it contains. For example, hypothesis  $r$  has a complexity of 1, and hypothesis  $r \wedge s \wedge w$  has a complexity of 3.

Based on the metrics for explanatory power and complexity, this study proposes a criterion to select good hypotheses, which is declared in Definition 6.

**Definition 6 (A criterion to select good hypotheses).** Given a set  $H$  of candidate hypotheses to explain a set  $F$  of facts,  $h \in H$  is considered a good hypothesis if it satisfies all the following conditions:

- The explanatory power of  $h$  is equal to or greater than a constant  $\lambda_1$ . The constant  $\lambda_1 = 0.5$  has been used in the experiments described in this article.
- The complexity of  $h$  is equal to or less than a constant  $\lambda_2$ . The constant<sup>1</sup>  $\lambda_2 = 5$  has been used in the experiments described in this article.
- The hypothesis  $h$  has the minimum complexity among all of the hypotheses that have the maximum explanatory power in  $H$ .

Examples 3 and 4 illustrate the application of Definition 6.

**Example 3.** Diseases manifest themselves through symptoms. Consider the following:

- Propositions ‘ $c$ : Disease is **cold**’, ‘ $p$ : Disease is **pneumonia**’, ‘ $r$ : Disease is **rhinitis**’, ‘ $f$ : Symptom is **fever**’, ‘ $h$ : Symptom is **headache**’, and ‘ $z$ : Symptom is **coryza**’;
- Theory  $T = \{ p \rightarrow f \wedge z \wedge h, c \rightarrow f \wedge z, r \rightarrow h \wedge z \}$ , the empty set  $C = \{ \}$  of accepted conditions and observed facts set  $F = \{ f, z, h \}$  (symptoms);
- A set of candidate hypotheses  $H = \{ p, c, r, p \wedge c, p \wedge r, c \wedge r, p \wedge c \wedge r \}$ .

Table 1 describes the explained facts, explanatory power and complexity of each candidate hypothesis  $h \in H$ .

**Table 1.** Explained facts, explanatory power and complexity of candidate hypothesis of the Example 3. The ‘√’ signals an explained fact.

Hypothesis	Explained facts			Explanatory power	Complexity
	$f$	$z$	$h$		
$p$	√	√	√	1	1
$c$	√	√		0.66	1
$r$		√	√	0.66	1
$p \wedge c$	√	√	√	1	2
$p \wedge r$	√	√	√	1	2
$c \wedge r$	√	√	√	1	2
$p \wedge c \wedge r$	√	√	√	1	3

All of the hypotheses have an explanatory power equal to or greater than  $\lambda_1 = 0.5$  and complexity equal to or less than  $\lambda_2 = 5$ . The hypotheses  $p, p \wedge c, p \wedge r, c \wedge r, p \wedge c \wedge r$  have explanatory power equal to 1, which is the maximum among all candidate

<sup>1</sup> The  $\lambda_1$  and  $\lambda_2$  values were chosen to coincide with human factors. Considering Miller experiment [12], human memory and human processing capacity is limited to  $7 \pm 2$  simultaneous elements, hence  $\lambda_2 = 5$ . Good hypotheses explain at least 50% of the facts, hence  $\lambda_1 = 0.5$ .

hypotheses. Among these hypotheses with maximum explanatory power, hypothesis  $p$  has the complexity equals to 1, which is the minimum among the hypotheses. Therefore,  $p$  is a good hypothesis according to Definition 6.

**Example 4.** Another example involving diseases and symptoms. Consider the following:

- Propositions ‘ $d$ : Disease is **dengue**’, ‘ $u$ : Disease is **flu**’, ‘ $b$ : Symptom is **breathlessness**’, ‘ $f$ : Symptom is **fever**’, ‘ $h$ : Symptom is **headache**’, ‘ $m$ : Symptom is **muscle pain**’, ‘ $r$ : Symptom is **red spots**’ and ‘ $s$ : Symptom is **sneezing**’;
- Theory  $T = \{ u \rightarrow f \wedge h \wedge m \wedge s, d \rightarrow f \wedge h \wedge m \wedge r \}$ , the empty set  $C = \{ \}$  of accepted conditions and observed facts set  $F = \{ f, h, m, b \}$  (symptoms);
- A set  $H = \{ u, d, u \wedge d \}$  of candidate hypotheses.

Table 2 describes the explained facts, explanatory power and complexity of each candidate hypothesis  $h \in H$ .

**Table 2.** Explained facts, explanatory power and complexity of candidate hypothesis of the Example 4. The ‘ $\checkmark$ ’ signals an explained fact.

Hypothesis	Explained facts				Explanatory power	Complexity
	$f$	$h$	$m$	$b$		
$U$	$\checkmark$	$\checkmark$	$\checkmark$		0.75	1
$D$	$\checkmark$	$\checkmark$	$\checkmark$		0.75	1
$u \wedge d$	$\checkmark$	$\checkmark$	$\checkmark$		0.75	2

All of the hypotheses have an explanatory power equals to 0.75 (i.e., explanatory power equal to or greater than  $\lambda_1 = 0.5$ ) and complexity equal to or less than  $\lambda_2 = 5$ . Hypotheses  $u$  and  $d$  have a complexity of 1, which is the minimum across all of the candidates. Therefore,  $u$  and  $d$  are good hypotheses according to Definition 6.

### 2.3 Pseudocode for the Algorithm Peirce

Figure 1 presents the pseudocode for the algorithm Peirce. The algorithm Peirce formulates hypotheses that comply with equations (5), (6), (7) and the criterion to select good hypotheses of the Definition 6. Synthetically, the algorithm Peirce formulates candidate hypotheses and stores them in set  $H$  (line 11). Next, the algorithm removes inconsistent hypotheses from  $H$  (line 12) and then selects and leaves only the good hypotheses in  $H$  (line 13). The details of the algorithm are described below.

The algorithm uses the Resolution rule of inference for PL sentences in CNF expressed with Horn Clauses [9]. Candidate hypotheses are hypotheses  $h$  that satisfy equation (6). To compute these hypotheses, the algorithm translates the set of sentences  $T \cup C \cup \neg F$  to CNF expressed with Horn Clauses (line 4) and applies the mechanism of resolution (line 5). The result of the resolution is stored in the data structure  $R$  (set of clauses). If  $R$  contains at least one empty clause, then  $T \cup C \models F$  and no hypotheses are formulated (lines 6 and 7). If  $R$  does not contain an empty

clause, then  $T \cup C \neq F$ , equation (5) is met and candidate hypotheses can be formulated.

At line 11 each clause in R presents the possibility of formulating a hypothesis. Because R is in CNF, negating each clause results in a candidate hypothesis. The algorithm `Formulate_Candidate_Hypotheses` (line 11), operates as follows: (1) the algorithm negates each of the  $m \geq 1$  clauses in R to obtain  $m$  first-candidates hypotheses  $h_1, h_2, h_3, \dots, h_m$ , (2) combines these  $m$  first-candidates hypotheses in pairs to obtain conjunctive hypotheses of the type  $h_i \wedge h_j$  ( $i \neq j$ ); and (3) combines the  $m$  first-candidates hypotheses three by three to obtain conjunctive hypotheses of the type  $h_i \wedge h_j \wedge h_k$  ( $i \neq j \neq k$ ); ... and then combines the  $m$  first-candidates hypotheses  $q$  by  $q$  to obtain conjunctive hypotheses of the type  $h_i \wedge h_j \wedge \dots \wedge h_q$  ( $i \neq j \neq \dots \neq q$ ), where  $q = \min(m, \lambda_2)$  and is  $\lambda_2$  the constant that defines the maximum complexity of the hypotheses ( $\lambda_2$  is defined in Definition 6).

```

Algorithm Peirce(T, C, F)
Input
Theory set T, accepted condition set C and facts set F
(specification is given in Definition 3).
Output
Hypotheses set H (specification is given in Definition 3).
1 {
2   if Consistent(T, C) then
3   {
4     R := Conjunctive_Normal_Form_Horn_Clauses(T, C, ¬F);
5     R := Resolution(R);
6     if R contains an empty clause then
7       write ("No hypotheses to formulate: T U C = F");
9     else
10    {
11      H := Formulate_Candidate_Hypotheses(R);
12      H := Remove_Inconsistent_Hypotheses(T, C, H);
13      H := Select_Good_Hypotheses(T, C, H, F);
14    }
15  }
16 else
17   write("Unable to formulate hypotheses: T U C = ⊥.");
18 }

```

Fig. 1. Algorithm Peirce.

At line 12, the algorithm `Remove_Inconsistent_Hypotheses` receives a set H of candidate hypotheses and removes from H hypotheses that do not satisfy  $T \cup C \cup \{h\} \neq \perp$  (conformity to equation (7)). The algorithm works as follows: For each  $h \in H$ : (1) the algorithm translates the sentences in the set  $T \cup C \cup \{h\}$  to CNF expressed with Horn Clauses, (2) applies the Resolution mechanism to this system of

sentences and (3) removes hypothesis  $h$  from  $H$  if the Resolution mechanism derives an empty clause.

The algorithm `Select_Good_Hypotheses` receives a set  $H$  of candidate hypotheses that are all consistent and then operates as follows: (1) it computes the explanatory power and complexity of each hypothesis  $h \in H$ , (2) removes all of the hypotheses  $h$  with explanatory power below some constant  $\lambda_1$  (0.5 in our experiments) or complexity above some constant  $\lambda_2$  (5 in our experiments), (3) computes set  $E$  with the hypotheses that have the maximum explanatory power in  $H$ , (4) computes set  $X$  with the hypotheses that have the minimum complexity in  $E$  and (5) returns set  $X$  as answer.

It can be proved that the Peirce algorithm computes three different types of solutions: (1)  $H = F$  when the theory and the accepted conditions does not allow Peirce algorithm to formulate explanatory hypotheses; (2)  $H = \{ \}$  when Peirce algorithm does not consider any hypotheses to be good, among the candidate hypotheses; (3)  $H$  contains at least one explanatory hypothesis; In this last type,  $H$  does not contain non-explanatory hypothesis.

Example 5 illustrates a run of the algorithm Peirce.

**Example 5.** This example illustrates the execution of the algorithm Peirce using the scenario and formalization from Example 2. Therefore, the algorithm Peirce receives as input the theory  $T = \{ r \rightarrow w, s \rightarrow w \}$ , the set of accepted conditions  $C = \{ t, t \rightarrow \neg s \}$  and the set of facts  $F = \{ w \}$ . Because  $T \cup C \not\models \perp$ , algorithm `Consistent(T, C)` returns the value true (line 2), and the data structure  $R$  is filled in with  $T \cup C \cup \neg F$  in CNF expressed with Horn Clauses. The following is then established:

- At line 4:  $R = \{ \{ \neg r, w \}, \{ \neg s, w \}, \{ t \}, \{ \neg t, \neg s \}, \{ \neg w \} \}$ ;
- At line 5 after Resolution:  $R = \{ \{ \neg r \}, \{ \neg s \} \}$ .

Because there are no empty clauses in  $R$  (test at line 6), the candidate hypotheses are formulated at line 11. Thus,  $H = \{ r, s, r \wedge s \}$  at line 11 after executing `Formulate_Candidate_Hypotheses`. Because  $T \cup C \cup \{ s \} \models \perp$  and  $T \cup C \cup \{ r \wedge s \} \models \perp$ , hypotheses  $s$  and  $r \wedge s$  are removed from  $H$  by the algorithm `Remove_Inconsistent_Hypotheses` (line 12), leaving  $H = \{ r \}$ . The hypothesis  $r$  has an explanatory power of 1, a complexity of 1 and the minimum complexity of all hypotheses with maximum explanatory power in  $H$  ( $r$  is the only hypothesis in  $H$ ), therefore the algorithm `Select_Good_Hypotheses` (line 13) selects  $r$  as a good hypothesis. The algorithm Peirce thus returns as answer  $H = \{ r \}$ .

In general, the complexity of logic-based abduction is NP-complete [6]. However, the algorithm Peirce has a running time  $O(n^{2+\lambda_2})$ . As  $\lambda_2$  is a constant, typically equals to 5, Peirce algorithm runs in polynomial time. This occurs by the following facts. The algorithm `Conjunctive_Normal_Form_Horn_Clauses` has running time  $O(n)$  because since every sentence of  $T$ ,  $C$  and  $F$  is restricted to HF form (Definition 4) they can be transformed directly into Horn Clauses in  $O(1)$ . The execution of the Resolution mechanism of the PL sentences in CNF with Horn Clauses can be done in  $O(n^2)$ . Thus, `Consistent` and `Resolution` algorithms have running time  $O(n^2)$ . The algorithm `Formulate_Candidate_Hypotheses` has a running time of  $O(n^{\lambda_2})$  because produces at most hypotheses combinations  $O(n^2) + O(n^3) + \dots + O(n^{\lambda_2})$ . The

algorithm `Remove_Inconsistent_Hypotheses` has a running time of  $O(n^{2+\lambda_2})$  because executes at most a constant amount of  $O(n^{\lambda_2})$  resolutions each of them in  $O(n^2)$ . The algorithm `Select_Good_Hypotheses` has a running time of  $O(n \log n)$ , to sort and select the set of hypotheses with minimal complexity among the hypotheses with maximum explanatory power.

### 3 Tendency of Solutions Computed by the Algorithm Peirce to Coincide with Abductive Reasoning Done by Humans

A study was realized to verify whether the abductive reasoning performed by humans tends to coincide with the solutions computed by the algorithm Peirce. The study was conducted using a questionnaire containing ten questions, with each question presenting an implicit description of a theory, observed facts and accepted conditions. The alternatives for each question present possible abductive hypotheses. Table 3 illustrates in the left column one question in the questionnaire.

**Table 3.** Example of a question used in the questionnaire. The left column describes the question itself, and the right column presents the corresponding formalization to the question and solution as computed by the algorithm Peirce.

Question	Formalization and solution computed by the algorithm Peirce
<p>Joshua is in the desert and sees something green in the distance. What would best explain what Joshua sees?</p> <p>a) I am convinced that it is a lawn.</p> <p>b) I am convinced that it is a cactus.</p> <p>c) I am convinced that it is a green flag.</p> <p>d) It could be either a cactus or a green flag.</p>	<p><b>Propositions:</b>                      ‘c: It is a <b>cactus</b>’, ‘d: It is a <b>desert</b>’, ‘f: It is a <b>green flag</b>’, ‘l: It is a <b>lawn</b>’, ‘s: Joshua <b>sees</b> something green’.</p> <p><math>T = \{ l \rightarrow s, c \rightarrow s, f \rightarrow s \}, C = \{ d, d \rightarrow \neg l \}, F = \{ s \}.</math></p> <p><b>Solution</b></p> <p>- After formulating candidate hypotheses (line 11):  <math>H = \{ l, c, f, l \wedge c, l \wedge f, c \wedge f, l \wedge c \wedge f \}.</math></p> <p>- After removing inconsistent hypotheses (line 12):  <math>H = \{ c, f, c \wedge f \}.</math></p> <p>- After selecting good hypotheses (line 13):  <math>H = \{ c, f \},</math> i.e., the alternative ‘d’ coincides with the solution of algorithm Peirce.</p>

The questionnaire, validated by a pilot-test with 25 individuals, was designed to be answered in 15 minutes. A total of 133 undergraduate and graduate students participated in the study. The profile of the participants showed a slight predominance of female individuals (53%) and ages ranging from 18 to 60 years, with an average and median close to 25 years.

Each participant’s answers to the questionnaire were computed, and one point was attributed to each answer on the questionnaire that coincided with a solution produced by the algorithm Peirce. The results showed an average of 86 answers coinciding with the algorithm Peirce and 47 that did not coincide.

The Chi-square ( $\chi^2$ ) test at 1% significance was used as a statistical measure of the significance with which the participants' answers coincided with solutions produced by the algorithm Peirce. For the studied population, the  $\chi^2$  test suggested that the coincidence between the participants' answers and the solutions computed by the algorithm Peirce was significant: ( $\chi^2(1) = 11.44, p\text{-value} = 0.001 < 0.01$ ).

## **4 Related Works**

Different approaches have been used to develop algorithms for abductive reasoning. Among the many contributions, there are proposals that use search techniques [15] and probabilistic reasoning over Bayesian Networks [7]. Logic approaches are based on two types of contributions: (1) proposal of new algorithms and (2) extension of traditional logical programming to process abductive reasoning problems.

Examples of type 1 contributions include [2] and [5]. Both of the proposals refer to abductive reasoning algorithms that operate with a THF reasoning framework (Definition 1). The main differences between these proposals and those of the present study are as follows: (1) They allow only one fact to be declared; (2) They do not allow define explicitly a set of accepted conditions; and (3) Semantic Tableaux is used in the proposal described in [2] instead of Resolution as the mechanism of inference.

Contributions of type 2 include Abductive Logic Programming (ALP) [11] and use the languages Prolog with Constraint Handling Rules (CHR) [1] [4]. The main differences between these proposals and those of the present study are as follows: (1) They operate with Predicate Logic; (2) They require special "abducible" predicates (possible hypotheses) to be declared; and (3) They dedicate little attention to criteria to select good abductive hypotheses.

Studies related to the one presented here, that address the selection of good hypotheses, include [8], [16] and, recently, [3]. This work differs from proposals [8] and [16] mainly by the proposed metrics for complexity and explanatory power of hypotheses.

## **5 Conclusions**

The abductive reasoning algorithm Peirce is distinct from other solutions mainly because it employs the TCHF reasoning framework and a simple criterion for selecting good hypotheses that consider quantitative metrics to define the explanatory power and complexity of the formulated hypotheses.

The TCHF reasoning framework has shown itself to be useful in organizing the elements that participate in abductive reasoning because it does not "contaminate" the theory with sentences that fundamentally do not belong to the theory. This framework provides an additional advantage because it explicitly exposes the conditions (contexts, circumstances, intentions etc.) under which the reasoning process is conducted, which is fundamental and frequent in the formulation of abductive reasoning.

The criteria for selecting good hypotheses are subjects of ongoing research. There is no consensus as to which criteria should be used and under which circumstances or in which domains they work. The criterion used by the algorithm Peirce, which is described in Definition 6, attempt to produce a simple algorithm that works in practice. Alternatives to Definition 6 exist and can be proposed.

The study that depicted the coincidence of the solutions computed by the algorithm Peirce to those derived through abductive reasoning performed by humans was not exhaustive because there is such a high number of domains, and it did not include the diversity and quantity of individuals. However, these results provided value suggesting that the abductive reasoning conducted by humans tends to coincide with the solutions computed by the algorithm Peirce.

## References

1. Alberti, M., Gavanelli, M., Lamma, E.: The CHR-based Implementation of the SCIFF Abductive System. *Fundamenta Informaticae* 124 (4), pp. 365–381 (2013)
2. Aliseda, A.: *Abductive Reasoning: Logical Investigations into Discovery and Explanation*. Springer, Netherlands (2006)
3. Caroprese, L., Trubitsyna, I., Truszczynski, M., Zumpano, E.: A Measure of Arbitrariness in Abductive Explanations. To appear in *Theory and Practice of Logic Programming*, 25 p (2014)
4. Christiansen, H.: Executable specifications for hypothesis-based reasoning with Prolog and Constraint Handling Rules. *Journal of Applied Logic* 7 (3), pp. 341–362 (2008)
5. Dillig, I., Dillig, T.: Explain: A Tool for Performing Abductive Inference. In Sharygina, N., Veith, H. (eds) *CAV 2013*. LNCS, vol. 8044, pp. 684–689. Springer, Heidelberg (2013)
6. Eiter, T., Gottlob, G.: The Complexity of Logic-Based Abduction. *Journal of the Association for Computing Machinery*, 42 (1), pp. 3–42 (1995)
7. Fortier, N., Sheppard, J., Strasser, S.: Abductive inference in Bayesian networks using distributed overlapping swarm intelligence. *Soft Computing Journal*, May 2014, pp. 1–21 (2014)
8. Harman, G. H.: The inference to the best explanation. *The Philosophical Review* 74 (1), pp. 88–95 (1965)
9. Howard, P.: *Introduction to Logic: Propositional Logic*. Prentice-Hall, New Jersey (1999)
10. Josephson, J. R., Josephson, S. G.: *Abductive Inference: Computation, Philosophy, Technology*. Cambridge University Press, Cambridge (1994)
11. Kakas, A. C., Kowalski, R. A., Toni, F.: Abductive Logic Programming. *Journal of Logic and Computation* 2 (6), pp. 719–770 (1995)
12. Mackenzie, I. S.: *Human-Computer Interaction: An Empirical Research Perspective*. Morgan Kaufmann, New York (2013)
13. Magnani, L.: *Abductive Cognition: The Epistemological and Eco-cognitive Dimensions of Hypothetical Reasoning*. Springer, Berlin (2009)
14. Peirce, C. S.: *Collected Papers of Charles Sanders Peirce*. Oxford University Press, London (1958)
15. Romdhane, L. B., Ayeb, B.: An Evolutionary Algorithm for Abductive Reasoning. *Journal of Experimental & Theoretical Artificial Intelligence* 23, pp. 529–544 (2011)

*Felipe Rodrigues, Carlos Eduardo A. Oliveira, and Osvaldo Luiz de Oliveira*

16. Thagard, P. R.: The Best Explanation: Criteria for Theory Choice. *The journal of philosophy* 75 (2), pp. 76–92 (1978)
17. Thagard, P. R.: Explanatory Coherence. *Behavioral and Brain Sciences* 12, pp. 435–502 (1989)

# Referências

- Aho, A. V., Lam, M. S., Sethi, R. & Ullman, J. D. (2006) *Compilers: Principles, Techniques, and Tools (2<sup>nd</sup> Edition)*. Boston: Addison-Wesley.
- Aliseda, A. (2006) *Abductive Reasoning: logical investigations into discovery and explanation*. The Netherlands: Springer.
- Callegari-Jacques, S. M. (2008) *Bioestatística: princípios e aplicações*. Porto Alegre: Artmed.
- Christiansen, H. (2008) Executable specifications for hypothesis-based reasoning with Prolog and Constraint Handling Rules. *Journal of Applied Logic*. Amsterdam: Elsevier BV, v. 7, n. 3, pp. 347-362.
- Dillig, I. & Dillig, T. (2013) Explain: a tool for performing abductive inference. In: Sharygina, N., Veith, H. (2013), eds. *CAV 2013*. LNCS, Heidelberg: Springer, v. 8044, pp. 684-689.
- Denecker, M. & Kakas, A. (2002) Abduction in Logic Programming. In: Kakas, A. C. & Sadri, F. (2002), eds. *Computational Logic: logic programming and beyond: essays in honour of Robert A. Kowalski Part I*. Berlin: Springer-Verlag, v. 2407, n. 1, pp. 402-436
- Doyle, A. C. (2007) *Sherlock Holmes: The Red Headed League*. Kettering: Crest Publishing House.
- Doyle, B. (2014) *Inference to the Best Explanation*. [Available at: [http://www.informationphilosopher.com/knowledge/best\\_explanation.html](http://www.informationphilosopher.com/knowledge/best_explanation.html)]. Last access at: March 8, 2015.
- Eiter, T. & Gottlob, G. (1995) The Complexity of Logic Based Abduction. *Journal of the Association for Computing Machinery*. New York: ACM, v. 42, n. 1, pp. 3-42.
- Folkler, A. L. E. (2002) *Automated Theorem Proving Resolution vs. Tableaux*. (Master Thesis). [Available at:

- [http://www.bth.se/fou/cuppsats.nsf/all/0599250d08df003dc1256b94006efa6a/\\$file/masterthesis100.pdf](http://www.bth.se/fou/cuppsats.nsf/all/0599250d08df003dc1256b94006efa6a/$file/masterthesis100.pdf)]. Last access at: March 8, 2015.
- Fonseca, R. (2009) *O Seminarista*. Rio de Janeiro: Agir.
- Hacking, I. (2001) *An Introduction to Probability and Inductive Logic*. Cambridge: Cambridge University Press.
- Harman, G. H. (1965) The inference to the best explanation. *The Philosophical Review*, v. 74, n. 1, pp. 88-95.
- Heit, E. (2007) What Is Induction and Why Study it?, In: Feeney, A. & Heit, E. (2007), eds., *Inductive Reasoning: experimental, developmental, and computational approaches (1<sup>st</sup> edition)*. New York: Cambridge University Press, pp. 1-24.
- Hopcroft, J. E., Motwani, R. & Ullman, J. D. (2006) *Introduction to Automata Theory, Languages, and Computation (3<sup>rd</sup> ed)*. Boston: Addison-Wesley.
- Howard, P. (1999) *Introduction to Logic: propositional logic*. New Jersey: Prentice-Hall.
- Josephson, J. R. & Josephson, S. G. (1994) *Abductive Inference: computation, philosophy, technology*. Cambridge: Cambridge University Press.
- Kakas, A. C., Kowalski, R. A. & Toni, F. (1995) Abductive logic programming. *Journal of Logic and Computation*, v. 2, n. 6, pp. 719-770.
- MacKenzie, I. S. (2013) *Human-computer Interaction: an empirical research perspective*. New York: Morgan Kaufmann.
- Magnani, L. (2009) Abductive Cognition: the epistemological and eco-cognitive dimensions of hypothetical reasoning. *Cognitive systems monographs*, v. 3. Berlin: Springer-Verlag.
- Mill, J. S. (1862) *A System of Logic, Ratiocinative and Inductive*. London: Savill and Edwards Printers. [Also available at: [http://books.google.com.br/books/about/A\\_System\\_of\\_Logic\\_Ratiocinative\\_and\\_Indu.html?id=y4MEAAAQAAJ](http://books.google.com.br/books/about/A_System_of_Logic_Ratiocinative_and_Indu.html?id=y4MEAAAQAAJ)]. Last access at: March 8, 2015.
- Nicoletti, M. C. (2010) *A Cartilha da Lógica*. São Carlos: EdUfscar.

- Nilsson, U. & Maluszynski, J. (1995) *Logic, Programming and Prolog*. New York: John Wiley & Sons Inc.
- Papert, S. A. (1980) *Mindstorms: children, computer, and powerful ideas*. New York: Basic Books.
- Peirce, C. S. (1867) On The Natural Classification of Arguments. *Proceedings of the American Academy of Arts and Sciences*, v. 7, pp. 261–287. Reprinted at *Collected Papers*, v. 2, paragraphs 461–516 and *Writings* v. 2, pp. 23–49.
- Peirce, C. S. (1934) *Collected Papers of Charles Sanders Peirce: volume 5, Pragmatism and Pragmaticism*. London: Oxford University Press.
- Peirce, C. S. (1958) *Collected Papers of Charles Sanders Peirce: volume 7, Science and Philosophy*. London: Oxford University Press.
- Romdhane, L. B. & Ayeb, B. (2011) An Evolutionary Algorithm For Abductive Reasoning. *Journal of Experimental & Theoretical Artificial Intelligence*, v. 23, n. 4, pp. 529-544.
- Thagard, P. R. (1978) The Best Explanation: criteria for theory choice. *The Journal of Philosophy*, v. 75, n. 2, pp. 76-92.
- Thagard, P. R. (1989) Explanatory Coherence. *Behavioral and Brain Sciences*, v. 12, n. 3, pp. 435-502.